

Materialien zur Vorlesung
Digitale Audioverarbeitung

Norman Hendrich

Universität Hamburg
Fachbereich Informatik
Vogt-Kölln-Str. 30
D 22 527 Hamburg
hendrich@informatik.uni-hamburg.de

Inhaltsverzeichnis

Einführung	1
Literatur	3
Studienprofil interaktive Medien	4
Aktuelle Nachrichten	5
Standort Hamburg	7
Trends	8
Gehör und Gehörwahrnehmung	10
Schall	10
Hörbereich	13
Aufbau des Gehörs	14
Lautheit, Tuningkurven	17
Klangfarben	20
Sprache	21
Lärm	24
Signalverarbeitung	26
Motivation	26
Zahlenfolgen, Impulsantwort	29
Fouriertransformation	30
Abtastung	31
Quantisierung	32
A/D-Wandlung	34
D/A-Wandlung	36
Dither	37
Audio-Signalverarbeitung	39
Direktform	40
Z-Transformation	41
Filter	42
FIR / IIR	44
Dynamikbearbeitung	46
Hall, Raumsimulation	48
Audioqualität	50
Transparenter Kanal	51
Vergleich CD, LP, Radio	52
Quantisierungseffekte	54
Audio-Hardware und -systeme	57
Soundblaster live, EMU10K1	58
Micronas MAS35X9F	59
Signalprozessoren	60
SHARC-Familie	61
SIMD-Befehlssätze	62
MMX	63
3Dnow!	67
ISSE und ISSE2	69
Benchmarks Pentium, Athlon	72
Compact Disc	74
CD Funktionsprinzip	75
Fehlerkorrektur (CD-ROM)	77
Datenformat CD-DA, CD-ROM	79
CD-ROM Dateisysteme	81
CD-R und CD-RW	82
DVD	86
DVD-Audio, SACD	88
DVD-Video	89
DVD-RAM, DVD-R	90

Dateiformate	92
Codec, SND, WAV	93
Sprach-Codex	95
Kompanierung	96
ADPCM-Kompression	98
SPDIF	99
ADAT	101
Psychoakustische Kodierung	102
MPEG-1	103
Filterbank	105
Encoder-Prinzip	107
Maskierung	108
Bitstrom	109
Pre-Echos und MPEG-2	110
LAME	112
Messungen	113
Streaming Media	114
Streaming im Internet: RTP, RTCP, RTSP	116
Windows Media	118
AVI und ASF	119
Windows Media Encoder	121
Real Networks, Liquid Audio, Quicktime	122
Napster und Gnutella	125
Audiovergleich	131
3D-Audio	132
Lokalisierung	133
HRTF	134
Mehrkanalton	136
Dolby Surround	137
Dolby Digital	138
DTS, MPEG-2, THX	139
Beispiel: Raum F334	140
DirectSound	142
EAX	145
Klangsynthese	148
Subtraktiv	150
Virtuell analog	152
Sampling	153
Additive Synthese	156
Frequenzmodulation	157
Physical Modeling	158
MIDI	160
MIDI Protokoll	162
Standard MIDI Files	165
General MIDI	166
Yamaha XG	167
Sequencer, Cubase VST	170
VST Mixer, Effekte, Instrumente	172
Wasserzeichen	174
Verschlüsselung	175
Beispiel MemoryStick	176
Audio-Wasserzeichen	180
SDMI, HackSDMI	183

"digitale Audioverarbeitung"

jede Form digitaler Verarbeitung von Audiodaten:

- Aufzeichnung und Wiedergabe: Sprache, Musik, Geräusche
- Datenspeicherung und Datenkompression
- Musik/Sprach-Synthesizer, virtuelle Tonstudios
- digitale Hörgeräte

Informatikaspekte:

- Datenformate, Datenkompression
- Datenübertragung, Streaming, insbesondere via Internet, QoS
- Audio-APIs (z.B. DirectX, JMF), 3D-Audio
- Suchmaschinen, Inhaltssuche, Tauschbörsen (Napster)
- digitales Copyright, Kopierschutz, Wasserzeichen
- MPEG-1, -2, -4, -7
- aktuelles Thema, große wirtschaftliche Bedeutung

Vorlesung

Digitale Audioverarbeitung

Norman Hendrich

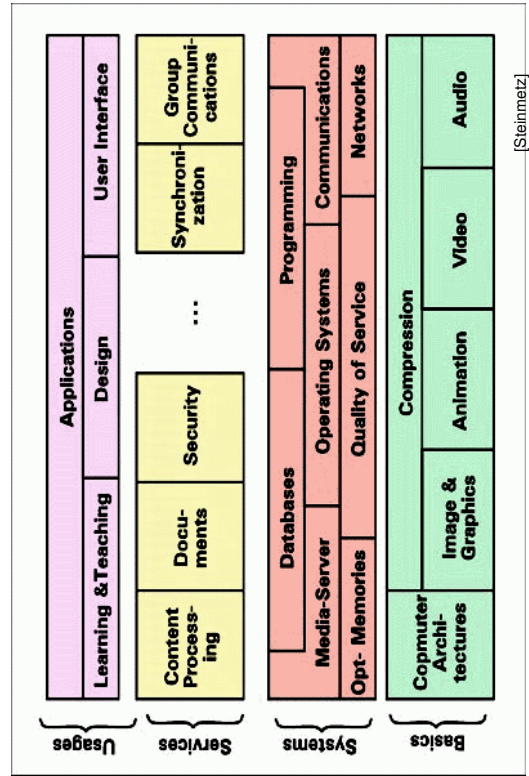
Universität Hamburg, Fachbereich Informatik, TECH
hendrich@informatik.uni-hamburg.de

tech-www.informatik.uni-hamburg.de/lehre/ws2000/v1-audioverarbeitung/

Terminvorschau: Do, 12-14, F-334

0	26.10	Einführung, Motivation
1	02.11	Grundlagen des Gehörs
2	09.11	digitale Signalverarbeitung
3	16.11	Audioqualität, "transparenter Kanal"
4	23.11	Signalverarbeitung, Audio-Hardware
5	30.11	Audio-Datenformate
6	07.12	CD / DVD / Minidisc / HD-Recording
7	14.12	psychoakustische Verfahren, MP3, AAC
8	21.12	Streaming, Napster
9	11.01	Surround-Verfahren, DirectX
10	18.01	Musiksynthese
11	25.01	MIDI, GM, Sequencer, virtuelles Studio
12	01.02	digitales Copyright, Wasserzeichen, Signaturen
13	08.02	Programmiersprachen für Musik, MPEG-4, MPEG-7

Medienverarbeitung



Audio: Trends

- Moore's Law: Rechenleistung steigt 50% / Jahr:
 - immer bessere DSP-Algorithmen möglich
- => Ersetzen externer HW-Geräte durch SW AC97-Codec statt Soundkarte
- => Highend-Soundkarte statt Tonstudio low-cost HD-Recording virtuelle Tonstudios, volle Audio/MIDI-Integration
- => völlig neue Möglichkeiten bessere Tonqualität (24 bit, 96 KHz), Surround 5.1, etc. 3D-Audio statt Stereo oder 5.1 Echtzeitmanipulation von Audiodaten (Beispiel: Autotune zur Gesangskorrektur)



Sie sehen gut aus, aber Ihr Gesang ist lausig?

ANTARES AutoTune und ATR-1: Perfekter Gesang aus der Box

Sie sehen gut aus, aber Ihr Gesang ist lausig? Kein Problem, denn mit ANTARES Auto Tune als Software oder dem ATR-1 Hardwaremodul wird Ihre Aufnahme brillant perfekt!

- 19" Gehäuse
- Bitformat 20 bit linear, 56 bit intern
- Samplingfrequenz 46,375 kHz
- 40-Wandlung 20 bit (0,03 dB Dynamic Range)
- 40-Wandlung 24 bit (0,05 dB Dynamic Range)
- Intelligenter MIDI-Sensierung
- Inputs und Outputs: XLR symmetrisch, Mikro symmetrisch/unsymmetrisch
- Display: 2 x 20 Zeichen LED, Korrektur-Indikator 10 x LED, Input-Level 6 x LED

Softwareversionen erhältlich für:

- Stand Alone: DM und VST für Mac
- Direct für IBM-compatible PCs

DM 1.999,-
 irreverentiale Preisgestaltung

ANTARES
 MULTIMEDIA SOFTWARE TECHNOLOGIES

Hyperactive Audioochhnik GmbH
 Schwarzwaldstr. 9
 63229 Bambrath
 Tel. 03 61 28 18 22 27
 Fax 03 61 28 18 22 28
 hyperactive@vml.de

Prüfung der VMD 4. V.
 Förderfähigkeit des VOT

AUTO-TUNE

[Echtzeitkorrektur von Gesang / Instrumenten - seit Q1/1999]

CD-Qualität

Consumer-Markt ist verwöhnt:

- analoge Datenträger (LP/Cassette) weitgehend durch CD verdrängt
 - bei verhältnismäßig geringen Kosten
- => Verbraucher erwarten "CD-Qualität"
- => neue Verfahren müssen sich an CDs messen
- sehr gute Klangqualität, 96 dB Dynamik, rauschfrei
 - Fehlerkorrektur, keine Aussetzer oder Knackser
 - Direktzugriff, Programmierbarkeit
 - einfaches Kopieren ("CDs brennen")
- Trend: Übergang zu DVD mit Surround-Audio

Universalinstrument PC

PC-Audio?

- Musik und Videos abspielen
 - Sprachausgabe / -eingabe
 - Streaming-Media, Telephonie
 - Modern-Funktionen
 - Musikaufnahme / -produktion
 - Musikinstrument, Synthesizer
 - 3D-Audio für virtual reality
- CD, DVD | MP3, AC3, MD, ...
- RealAudio, MS media
 AC97, AMR, CMR
- Cubase, Logic, ...
 z.B. VST-Plugins
- Spiele, VR-Anwendungen
- => höchst unterschiedliche Anforderungen
- => Bandbreite KB/s bis GB/s, Rechenleistung bis GFLOPS

Literatur: Bücher

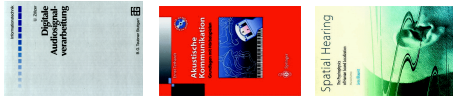
Udo Zölzer, digitale Audiosignalverarbeitung
Teubner, ISBN 3-519-06180-5

Ernst Terhardt, Akustische Kommunikation
Springer, ISBN 3-540-63408-8

Jens Blauert, Spatial Hearing
MIT-Press, ISBN 0-262-02413-6

Peter Gerdson, Peter Kröger, Digitale Signalverarbeitung
in der Nachrichtenübertragung, Springer, ISBN 3-540-61194-0

Klaus von der Heide, Matlab-Skripte zu
Digitale Signalverarbeitung / Nachrichtenübertragung
tech-www.informatik.uni-hamburg.de/lehre/



Digitale Audioverarbeitung | WS 2000 | 18.205

Literatur: WWW

tech-www.informatik.uni-hamburg.de/lehre/ws2000/vi-audioverarbeitung/
tech-www.informatik.uni-hamburg.de/lehre/medien-audio/

www.doctoraudio.com
www.synthzone.com
(interessante Firmen-Links)

www.google.com
www.altavista.com
www.napster.com
(Suchmaschine, 2nd generation)
(Mediensuche, bisher aber kaum brauchbar)
(Internet Tauschbörse)

Digitale Audioverarbeitung | WS 2000 | 18.205

Literatur: Paper

Birger Kollmeier., Kursunterlagen Audiologie und Signalverarbeitung
www.physik.uni-oldenburg.de/Docs/medi/download/docs/lehre/index.html

Creative / Areal Whitepaper zu EAX/A3D
Microsoft Whitepaper und MSDN Doku zu DirectX und Windows Media
Dolby Labs AC3 Dokumentation
Fraunhofer IIG MP3 Dokumentation & Quellcode

- siehe Liste auf tech-www

Digitale Audioverarbeitung | WS 2000 | 18.205

Literatur: Skript

das Skript entsteht "just in time"

- Beamer, keine Ausdrucke der Folien vorhanden
- aktuelle Fotienkopien jeweils auf dem Webserver
tech-www.informatik.uni-hamburg.de/lehre/ws2000/vi-audioverarbeitung/
- Anregungen, Vorschläge, Korrekturen bitte an
hendrich@informatik.uni-hamburg.de
- Vervielfältigung via ZVV bei Bedarf/Interesse möglich
- aber vermutlich erst am Semesterende

Digitale Audioverarbeitung | WS 2000 | 18.205

Studienprofil: interaktive Medien

Der Rahmenstudienplan wird erweitert um ein fünftes Studienprofil "interaktive Medien" mit folgenden Angaben:

[... Ergänzungsfächer, Grundlagenfächer, Schwerpunkte, usw.]

... schlägt das Curricularteam die Einrichtung vorerst nur eines Schwerpunkts im Profil vor, um die Anlaufschwierigkeiten zu minimieren. Eine fachliche Ausdifferenzierung kann erfolgen, wenn eine gewisse Konsolidierung des Angebots erreicht wurde.

... Studienprofile dienen der Orientierung der Studierenden.

Das Curricularteam sieht daher die Notwendigkeit zur Etablierung eines überschaubaren Studienangebots, das dennoch die fachliche Breite des Faches angemessen widerspiegelt. Gegenüber den derzeitigen Angeboten in den bestehenden Schwerpunkten, sind für die Zwecke des neuen Studienprofils vorrangig Lehrveranstaltungen mit großer fachlicher Breite, aber geringer Tiefe zu konzipieren.

[FB Informatik, 9/2000]

Studienprofil: interaktive Medien

CGB, 1 aus GBI, DOS
 STE, 1 aus DIS, SNN, VSS, WBS
 DKR
 LOS, PNL, 1 aus AUK, STH

Interaktionsdesign
 Entwurf von Hypermediasystemen
 Computergestützte kooperative Arbeit
 Bildverarbeitung
 => **Audioverarbeitung**
 Textanalyse
 Virtuelle Realität
 Verteilte Systeme für interaktive Medien
 Netzsicherheit, Kryptologie
 Multimedia-Datenbanken
 => **Medientechnik**

große Breite, geringe Tiefe :-)

oberflächlich:

- Aufbau des Gehörs, Gehörwahrnehmung
- Signalverarbeitung (Grundlagen, Mathematik)
- Datenformate, Datenspeicherung
- psychoakustische Verfahren
- eCommeroe, Copyright, Signaturen, Wasserzeichen

Themen mit einiger Tiefe:

- Signalverarbeitung (typ. Audio-Algorithmen)
- Rechnerarchitektur für Medienverarbeitung
- Musiksynthese
- Studientechnik, MIDI, virtuelle Studios

Musikproduktion . . .

Portrait of someone else

Das Demo des Siegetitels, das Sie in MP3-Qualität auf CD-Track 01 hören können, ist eine reine MIDI-Produktion, bei der Emagics Logic Audio Macintosh als Sequenzer-Zentrale zum Einsatz kam.

Besonders interessant: Marc Müller spielte alle MIDI-Tracks – also nicht nur Akkorde und Melodien, sondern auch Drums und Percussion – mit einer Gitarre als MIDI-Controller ein. Seine Godin Nylon-String-Gitarre ist zu diesem Zweck mit einem integrierten „MIDI-Pickup“ ausgestattet, der alle gespielten Note-Events an den Eingang eines MIDI-Interface übermitteln kann.

- Gitarre plus Sampler
- diverse Effektgeräte
- Mischung MIDI/Audiodaten [Keyboards 10/2000]

Die meisten Sounds des Backings (Percussion, Drums und Synth-Sounds) kommen im Demo als Samples von einem Akai S-3000. Als einziger weiterer MIDI-Klangerzeuger ist ein Alesis DM 5 Drumcomputer zu hören, der eine Snare und einen Kick-Sound zur Dopplung des Bassdrum-Samples beisteuert.

Die Gitarren-Linien spielte Marc ebenfalls auf seiner Godin ein, diesmal aber natürlich nicht in Form von MIDI-Daten: Das Ausgangssignal des Audio-Pick-Ups wurde per Line-In direkt auf einen DAT-Recorder aufgenommen. Danach suchte Marc auf dem DAT-Tape die besten Gitarren-Phrasen aus, reichte den So-land-Effektprozessor DEP-5 an, samplete das Ganze wieder mit dem S-3000 und triggerte die fertigen Gitarren-Samples schließlich durch eine Logic-MIDI-Spur an.

Auch der Mixdown gelang mit relativ wenig Aufwand: Die Signale aus Sampler und DEP-5

Napster



Über 6.7 Millionen Napster-User allein in den USA

[06.10.2000 10:13]

Die Musik-Tauschbörse Napster[1] entwickelt sich derzeit zur am schnellsten jemals über das Internet verbreiteten Software. Wie die Marktforscher von MediaMatrix[2] jetzt bekannt geben, ist die Zahl der User, die Napster von zu Hause aus benutzen, in den USA von 1,1 Millionen im Februar dieses Jahres auf über 6,7 Millionen im September gestiegen. Damit sind von den insgesamt knapp 80 Millionen US-Amerikanern, die von zu Hause aus im Internet surfen, 8,5 Prozent Napster-User. Und vom Arbeitsplatz aus frequentieren den Dienst zusätzlich über 1,1 Millionen Surfer. Welche Reichweite Napster weltweit hat, teilen die Marktforscher nicht mit.

Der Spieß für die Napster-User könnte aber bald vorbei sein. Derzeit kämpft die Tauschbörse von einem amerikanischen Berufungsgericht ums Überleben[3]. Die Vereinigung der amerikanischen Musikindustrie hatte gegen Napster wegen angeblicher Verstöße gegen Urheberrechtsbestimmungen geklagt. (axv[4](c,t))

URL dieses Artikels:

http://www.heise.de/newsticker/data/axv-06.10.00-0007

Links in diesem Artikel:

- [1] <http://www.napster.com>
- [2] <http://www.mmxl.com>
- [3] <http://www.heise.de/newsticker/data/jk-03.10.00-0017>
- [4] <mailto:axv@ct.heise.de>

=> zukünftiges Vertriebsmodell für Medien und Software ?!

Digitale Audioverarbeitung | WS 2000 | 18.205

Napster-Geschäftsmodell



Musikindustrie hält nichts von Napsters Geschäftsmodellen

[02.10.2000 17:02]

Der Prozess gegen die Musik-Tauschbörse Napster[1] wird am heutigen Montag am Berufungsgericht des neunten US-Gerichtsbezirks in San Francisco fortgesetzt. Nun soll endgültig über das Schicksal Napsters entschieden werden. Die Richter können das Verfahren aber auch an die vorherige Instanz zurückverweisen, in der Bezirks-Richterin Marilyn Hall Patel gegen Napster eine einstweilige Verfügung[2] ausgesprochen hatte. Diese war erst in letzter Minute aufgehoben[3] worden. Der Verfügung zufolge hätte Napster seinen Dienst einstellen müssen, Amerikanische Beobachter halten es jedoch für wahrscheinlicher, dass die Richter der Berufungsinstanz den Fall selbst entscheiden.

Die Vereinigung der amerikanischen Musikindustrie (RIAA[4]) will den Fall gegen Napster offensichtlich auf jeden Fall durchziehen. Angebote Napsters hatte die RIAA stets zurückgewiesen. So etwa auch die Vorstellung von Napster-Chef Hank Barry, dass, wenn jeder User etwa pauschal 4,85 US-Dollar monatlich für die Nutzung von Napster zahle, die RIAA seiner Meinung nach bis zu 500 Millionen US-Dollar jährlich an Lizenzgebühren erhalten könnte. Barry schließt daraus, dass es der RIAA bei dem Prozess nicht allein um Geld gehe, sondern um die vollständige Kontrolle des Musikgeschäfts.

Die Argumente in dem seit längerem laufenden Verfahren haben sich bis heute kaum geändert: Napster besteht darauf, selbst keine Verletzung des Urheberrechts begangen zu haben und auch seine Nutzer davor zu warnen. Auch das Käuerverhalten habe sich durch die Online-Tauschbörse kaum geändert; diese Ansicht wird dabei sogar teilweise durch Studien der Musikindustrie gestützt. Die Plattenfirmen und einige Künstler sind anderer Meinung: Sie gehen mit zunehmend härteren Maßnahmen gegen die Musikangebote im Internet und den damit angeblich einhergehenden Umsatzschwund vor.

• Kavaliersdelikt Raubkopieren vs. legale Tauschbörse ?

Digitale Audioverarbeitung | WS 2000 | 18.205

Napster



Hintergrund: Das Establishment frisst den "Revolutzer"

[31.10.2000 19:17]

"Wir machen es den Leuten wirklich schwer, legal an Musik im Netz zu kommen", sagte kürzlich Andreas Schmidt, Chef der Bertelsmann eCommerce Group. Untersuchungen hätten belegt, dass zum Beispiel 76 Prozent der Napster-Nutzer für Musikstücke zahlen würden, wenn sie legal und in guter Qualität angeboten werden würde. "Der Austausch digitaler Musikstücke ist nicht aufzuhalten", sagt Schmidt. Und wenn man es nicht selbst hinkommt, dann beteiligt man sich eben an dem Vorreiter der Tauschbörsen im Internet – oder übernimmt ihn gleich ganz. Die so genannte "strategische Allianz" zwischen Bertelsmann und Napster, die am heutigen Dienstag verkündet wurde[1], ist daher nur konsequent.

Bertelsmann bekommt Zugang zu einem eingeführten Dienst, Napster einen finanzstarken Partner, der gleich noch ein Musiklabel in die neue Ene mit einbringt. Und die derzeit weltweit 37 Millionen registrierten Napster-Nutzer sind allein ein Wert, den man gar nicht hoch genug schätzen kann – allein für diese Daten hätte sich ein Vielfaches des angeblichen Kaufpreises, wohl ein zweistelliger Millionenbetrag, gelohnt. Doch dazu kommt noch, dass es sich um die aktivsten Nutzer handeln dürfte, die eine ganz besonders wertvolle Zielgruppe darstellen.

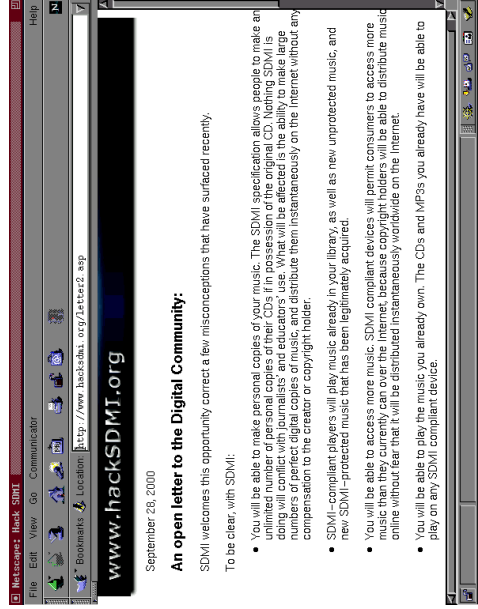
Stufenmodell

Wie bei allen Internet-Anwendungen, so wird nun auch bei Napster die nächste Stufe des Geschäftsmodells eingeführt: Aus der kostenlosen Tauschbörse wird ein Abonnement-Modell. Offiziell wird es damit begründet, dass Urheberrechte damit abgegolten werden. Doch andererseits entspricht es exakt dem Plan jedes kommerziellen Internet-Geschäfts: Erst einmal Marktanteile besetzen und einen Kundenstamm aufbauen, auch unter Inkaufnahme hoher Anlaufverluste durch kostenlose Angebote. Und dann erst kommt das eigentliche Geschäftsmodell, mit dem irgendwann auch Gewinne eingefahren werden sollen.

=> Abonnement-Modell

Digitale Audioverarbeitung | WS 2000 | 18.205

hacksdmi.org



• alle Verfahren bereits geknackt ?!

Digitale Audioverarbeitung | WS 2000 | 18.205

RealAudio und Sony vs. MP3



Real und Sony mit neuem Audio-Format fürs Internet

[23.10.2000 16:08]
Real Networks[1] veröffentlicht in Zusammenarbeit mit Sony ein **neues Format**[2] für Internet-Audio. RealAudio 8 soll ältere Formate des Streaming-Media-Marktführers Real Networks ablösen und gleichzeitig wesentlich bessere Klangqualität bieten. Dazu setzt Real erstmals nicht auf eine reine Eigenentwicklung, sondern auf den von **Sony**[3] entwickelten ATRAC3-Audiostandard. Damit feiert das bisher für die Minidisk eingesetzte Audiiformat seine Premiere als Audio-Format im Internet.

Nach Aussagen von Real Networks soll RealAudio 8 bereits bei Bitraten von 64 kBit/s kaum von CD-Qualität zu unterscheiden sein. Ob das neue Format halt, was Real Networks verspricht, kann man nach Download eines Plug-Ins für den aktuellen Real Player **3 selbst beurteilen**[4]. Ein erster, kurzer Hörtest in der Redaktion erbrachte zumindest durchaus positive Eindrücke. **(vza[s](c)t)**

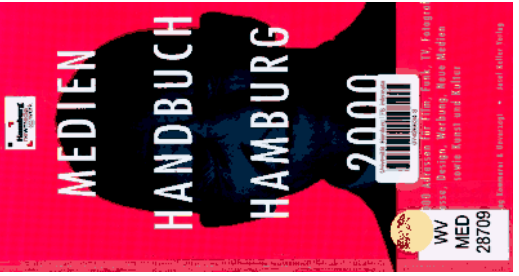
URL dieses Artikels:
<http://www.heise.de/newsticker/data/vza-23.10.00-000/>

Links in diesem Artikel:

- [1] <http://www.real.com>
- [2] http://www.realnetworks.com/feature/index_1023000.html?src=mmfm
- [3] <http://www.sony.com>
- [4] <http://realguide.real.com/rab/>
- [5] <mailto:vza@ct.heise.de>

- gleiches Konzept wie MP3, AC3, ... aber schlechter dokumentiert

Standort Hamburg: Medien-Handbuch



Medienhandbuch Hamburg 2000

"über 10.000 Adressen für Film, Funk, ..."
Rubriken: 3D-Modelling, Abspernung, Archive, Art Consulting, Artisten, Audiovision, ...
Casting, Computer HW/SW für den Medienbereich, ..., Fotografen, ... Untertitelung, ... Webmasters

"Medien"-Studieren in Hamburg:

- Uni-Hamburg: div. Fachbereiche, FB18
- TUHH
- FH Hamburg
- FH Wedel
- HfbK
- HFMT (Musikhochschule)

Zukunftspreis für MP3



MP3-Erfinder erhalten Deutschen Zukunftspreis

[20.10.2000 09:51]

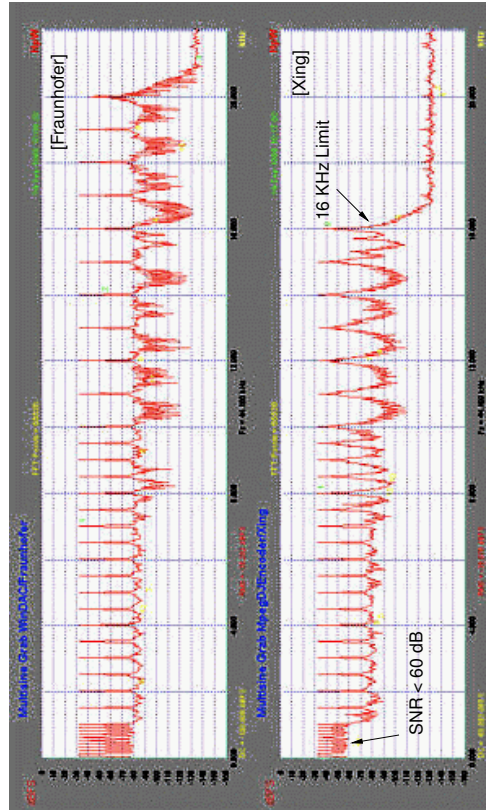
Der **Preis des Bundespräsidenten für Technik und Innovation**, kurz unter dem Label **Deutscher Zukunftspreis**[1] geführt, geht in diesem Jahr an die Erfinder der Audio-Kompressionstechnik MP3. **Ausgezeichnet**[2] wurden drei **Mitarbeiter**[3] des Fraunhofer-Instituts für Integrierte Schaltungen **IS-A[4]**: Prof.-Dr. Prof. Kai-Inhvez Brandenburg Dipl.-Ing. Bernhard Grill und Dipl.-Ing. Harald Popp.

Zur Begründung der Preis-Verleihung erklärte die Jury: "Mit dieser Innovation wird über die Codierung akustischer Signale und die Verdichtung des Datenstromes eine Komprimierung von Musik ohne Qualitätsverlust erzielt. MP3 wurde Weltstandard und hat insbesondere die Wettbewerbsfähigkeit der einschlägigen deutschen Industrie gestärkt. Bundespräsident Johannes Rau verbindet mit dem Preis die Hoffnung, die besondere Bedeutung von Spitzenleistungen in Wissenschaft und Technik für eine erfolgreiche Weiterentwicklung Deutschlands stärker ins öffentliche Bewusstsein zu bringen. Die Ergebnisse von Forschung und Entwicklung entscheiden maßgeblich über unsere Zukunft. Sie sind der Motor für gesellschaftliche, ökonomische und ökologische Erneuerung und - damit nicht zuletzt - für die individuellen Perspektiven der Menschen in unserem Land."

Neben der Komprimierungsrate und der Audio-Qualität der MP3-Dateien hob Bernhard Grill ein Prinzip des MP3-Standards als bemerkenswert hervor: "Der Empfänger [Decoder] ist festgelegt, aber man kann den Sender [Encoder] noch lange nach dieser Festlegung verbessern, wodurch wir auch nach der Standardisierung von MP3 gewaltige Fortschritte erzielen konnten."

- engültiger Durchbruch für psychoakustische Verfahren

MP3: zwei Encoder ...



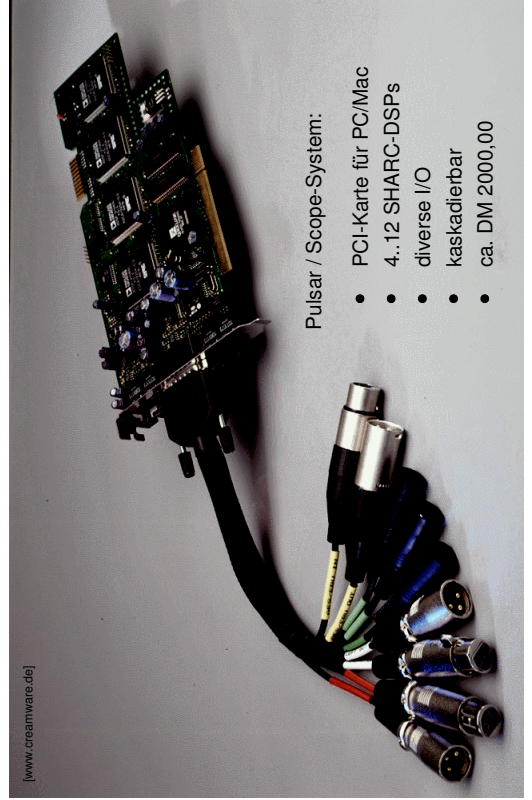
- Eingangssignal mit 44 Sinustönen ... [ct 21/98 242]

Standort Hamburg: Emagic Werbung



Digitale Audioverarbeitung | WS 2000 | 18.205

Trend: Parallelrechner



Digitale Audioverarbeitung | WS 2000 | 18.205

Standort Hamburg: NDR



- plus diverse Privatsender in/um Hamburg
- Studio Hamburg, ...

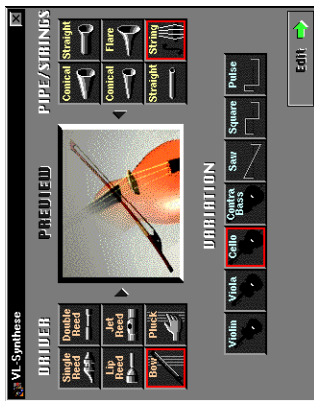
Digitale Audioverarbeitung | WS 2000 | 18.205

Standort Hamburg: Steinberg



Digitale Audioverarbeitung | WS 2000 | 18.205

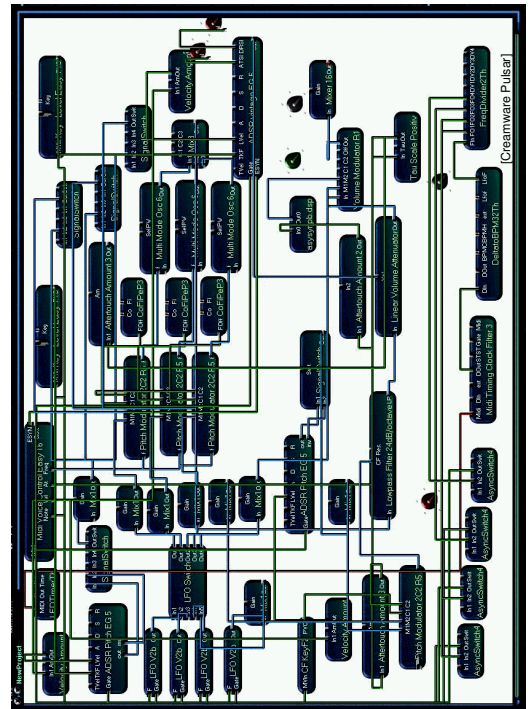
Trend: physical modeling



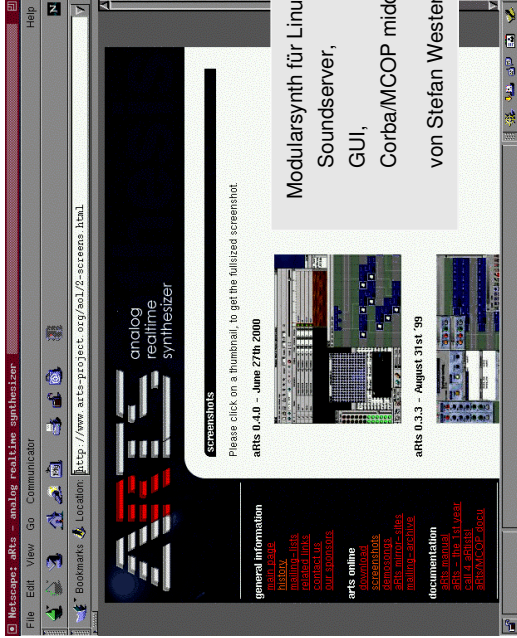
Beispiel:
Saiteninstrument
Resonanzen wie Cello
Anregung per Streichen
plus Effekte
alle Kombinationen möglich!
nicht alle sinnvoll ...

- mathematisches Modell von Musikinstrumenten
z.B. schwingende Saite, Luftsäule,
- Lösung der Differentialgleichungen (in Echtzeit...)
- Yamaha VL-Synthesizer, mittlerweile auch in Software

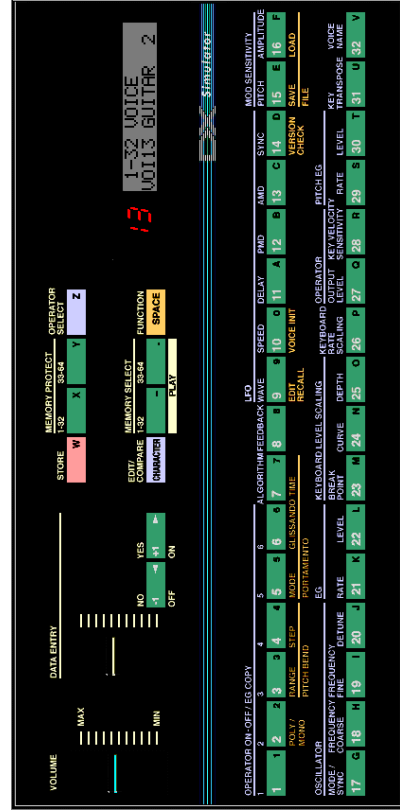
Trend: modulare Synthesizer



Trend: modulare Synthesizer: aRts



Trend: photorealistische Oberflächen...



- GUI als Original-Bedienfeld des DX7 Synthesizers
- inklusive Folientaster, Siebensegmentanzeigen, ...
- Ergonomie ?!

Trend: Software-Synthesizer auf PC/Mac



"Ideen können schnell über Hardware hinauswachsen."
Pentium/Athlon statt separater Audiohardware...

© IBM, Microsoft, Intel, AMD, Creative

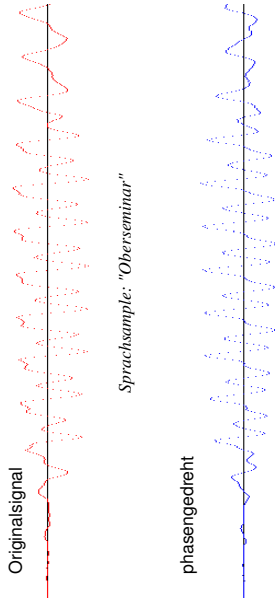
[Native Instruments]

Trend: 3D-Audio

räumliche Abbildung von Schallquellen:

- besser als Stereo oder Surround-Systeme
- für Spiele oder VR-Anwendungen
- für Filme
- Bedeutung für Musik ?!
- HRTF-Modellierung, Lautstärke bis Doppler-Effekt
- Problematik Kopfhörer / Lautsprecher
- Literatur zu 3D-Audio:
 - www.sensaura.com
 - AD / Sensaura Präsentation Intel Developer Forum 2000
 - www.sblive.com (Creative Labs EAX)
 - www.microsoft.com/directx

Trend: psychoakustische Verfahren



Demo: Hilbertfilter

- Hilbertfilter verändert gezielt die Phase eines Signals
- Ohr unempfindlich gegen Phaseninformation
- verändertes Signal klingt (vollkommen) gleich
- perceptual audio

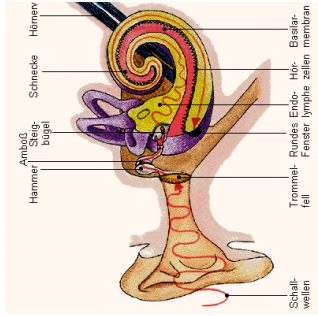
Audiodemo: Pandora

Zeitregen von Musiksignalen:

- mathematisch als Entfaltung
- aber unbekannte Eigenschaften der Quellen (z.B. Sänger)
- näherungsweise Lösung mit DSP und NN
- Pandora-Software extrahiert / verbessert Gesangsstimmen
- in beliebigem Audio-Mix
- Abtrennung von perkussivem Material
- Abtrennung der Stimm-Frequenzbereiche
- Audiovorführung aus [Keys CD 10/97]
- Pandora-Software derzeit nicht mehr als Produkt erhältlich

Gehörwahrnehmung: Agenda

- Schall und Schallausbreitung
- Aufbau des Gehörs
- Außenohr, Mittelohr, Innenohr
- Frequenzanalyse in der Cochlea
- Hörschwelle, Hörbereich, Lautheit
- Tonhöhe, Lautstärke, Klangfarbe
- Klangfarbe und Spektren
- Lärmempfindung, Schädigungen
- Sprache, Vokalartikulation



Motivation

wozu die Gehörwahrnehmung untersuchen?

- Forschung / Erkenntnisgewinn
- möglichst optimale Verfahren zur:
 - Aufzeichnung / Speicherung / Wiedergabe von Audio
 - so gut wie nötig - aber nicht besser (!)
- psychoakustische Verfahren, etwa MP3-Kompression
- bessere Algorithmen für Musik und Musikproduktion
- Raumsimulation für Spiele oder virtuelle Realität
- möglichst effiziente (low-level) Sprachverarbeitung
- medizinische Anwendungen, vor allem Hörgeräte

Schall

physikalischer Schallbegriff:

- alle elastischen Schwingungen eines Materials
- im allg. Fall longitudinal und transversal
- aber in Gasen (Luft) nur longitudinal (=Druckschwingungen)
- Schallgeschwindigkeit ca. 340 m/s in Luft
- Schallintensität lokale Druckänderung
 - Luftdruck auf Meereshöhe 1 bar = 10.000 Pa
 - Schmerzschwelle 1 mbar = 10 Pa
 - wahrnehmbare Druckänderung 10E-9 bar = 0.00001 Pa
- Ohr hat optimale Empfindlichkeit
- weitere Steigerung: Wahrnehmen der Molekularbewegung

Schall: Leistung, dB

- Kenngrößen des Schallfelds: Druckamplitude P bzw. Intensität I
- Gesamtleistung: Integral der Intensität über eine Oberfläche
- Lautstärke (Intensität): dB

$$L / \text{dB} \quad := 10 \log_{10}(I / I_{\text{ref}})$$

$$\quad := 20 \log_{10}(P / P_{\text{ref}})$$

$$P_{\text{ref}} \quad := 0.00002 \text{ Pa}$$

- alternativ: dB als Maß für das Verhältnis (p_1/p_2)

20 dB	zehnfacher Schalldruck
6 dB	doppelter Schalldruck
-6 dB	halber Schalldruck

Quelle	Leistung (typ.)
Sprache	10 μ W
Geige	1 mW
Lautsprecher	100 W

Schall: Phon

Phon: hörbezogene Einheit der Schallintensität
 Intensität eines gleichlaut empfundenen Referenztons
 logarithmische Skala notwendig (13 Größenordnungen)

$$I/\text{phon} := 10 * \log_{10}(I / I_0)$$

I_0 := Intensität der Hörschwelle bei 1000Hz
 := 10E-16 W/cm²

Quelle	Phon
Hörschwelle	0 (I = I ₀)
Flüstern	10 (I = 10 * I ₀)
Sprache	50 (I = 100.000 * I ₀)
laute Radiomusik	80 (I = 1.000.000.000 * I ₀)
Disco, Preßlufthammer	130 (Schmerzschwelle)

Koordinatensystem

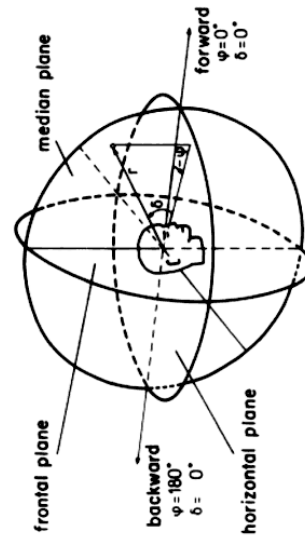


Figure 1.4 A head-related system of coordinates used in auditory experiments; r is distance, φ is the azimuth, and δ is the elevation.

- sphärische Koordinaten sinnvoll
- bei Bedarf Umrechnung Welt/Kopf-Koordinaten

Schall: Nah- und Fernfeld

[Blauert]

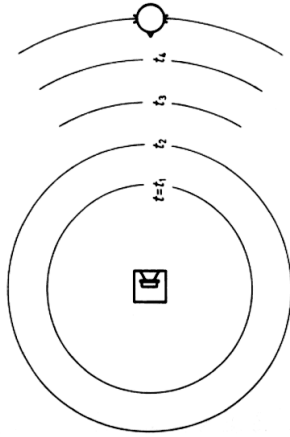


Figure 1.14 Any pulsating sound source that is small compared with the wavelength generates spherical waves at a moderate distance. As the distance is increased, the spherical waves become more and more plane.

- relative Größe / Entfernung von Quelle und Hörer

Schall: Nah- und Fernfeld

[Blauert]

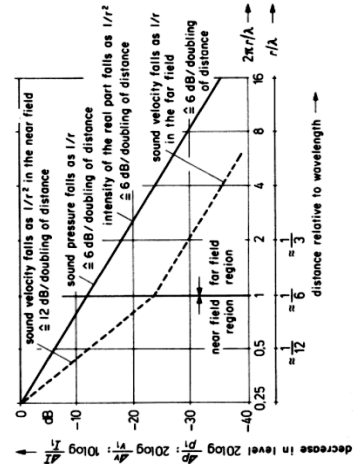


Figure 1.13 Pressure, velocity, and intensity of the sound field near to and distant from a spherical radiator of the zeroth order.

- typischer Abfall mit 6dB bzw. 12 dB / Entfernung

Schall: Beugung an einer Kugel

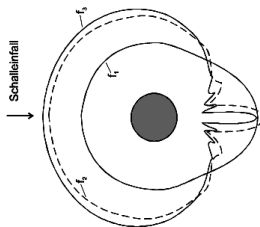


Abbildung 2.4: Beugungsbild bei Schallbeugung an einer Kugel für unterschiedliche Frequenzen, wobei $\theta_1 < \theta_2 < \theta_3$ [Kollmeier]

Bei niedrigen Frequenzen stellt die Kugel fast kein Hindernis dar, während bei hohen Frequenzen der Grenzfali der **geometrischen Akustik** bereits angenähert ist, d. h. eine starke akustische Schattenwirkung und eine geringe seitliche Streuung. Interessanterweise entsteht immer auf der genau der Schalleinfallrichtung abgekehrten Seite („auf der akustischen Achse“) ein Schalldruckmaximum, das man sich nach dem Babinetischen

Raumwahrnehmung

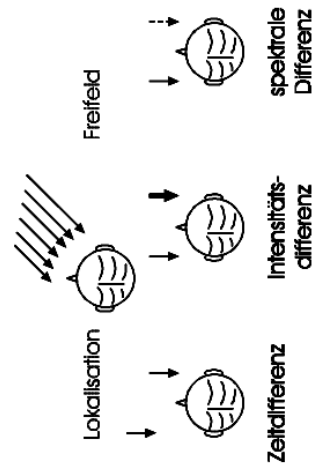


Abbildung 5.11: Physikalische Merkmale für die Lokalisation [Kollmeier]

- für 3D-Audio müssen alle Effekte beachtet werden
- zusätzlich Doppler-Effekt für bewegte Quellen

Raumwahrnehmung

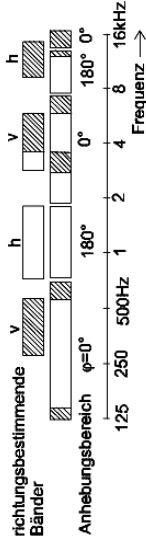


Abbildung 2.2: Richtungsbestimmende Bänder nach Blauert (aus Blauert, J.: Räumliches Hören. Nachschrift. S. Hirzel Verlag, Stuttgart 1985) Die erste Zeile gibt den Frequenzbereich an, bei dem in psychoakustischen Tests die Anwesenheit (schraffiert) oder Abwesenheit (nicht schraffiert) einer spektralen Anregung zu einer Lokalisation nach vorne („v“) bzw. hinten („h“) führt. Die zweite Spalte zeigt analoge Auswertungen aus Übertragungsfunktionen für Schalleintfall von vorn ($\phi = 0^\circ$) bzw. von hinten ($\phi = 180^\circ$). [Kollmeier]

Raumwahrnehmung: Cone of confusion

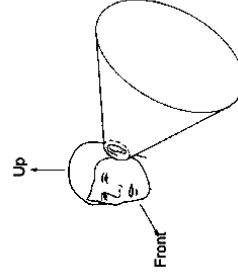
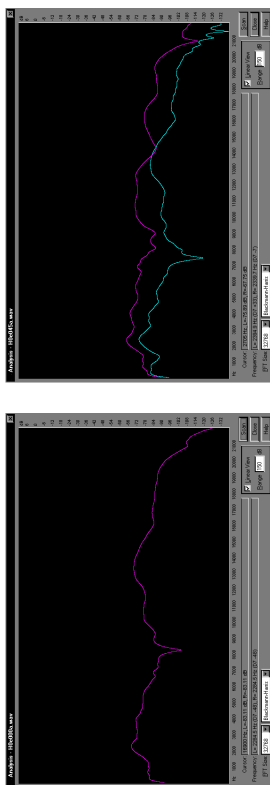


Abbildung 2.1: Cone of confusion, d.h. Fläche im dreidimensionalen Raum, auf der sämtliche Schalleintfallrichtungen dieselbe interaurale Zeit- und Pegeldifferenz aufweisen (aus Moore, B. C. J.: Perceptual Consequences of Cochlear Damage. Oxford University Press, Oxford 1995) [Kollmeier]

Raumwahrnehmung: HRTF



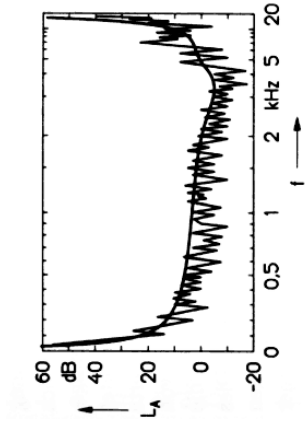
45 Grad seitlich

von vorne

Head-related transfer function

- individuelle Kopf (Außenohr)- Übertragungsfunktion
- mißt Intensität des Schalls am Trommelfell, links/rechts separat
- außerordentlich komplexe Winkel/Frequenzabhängigkeit
- Beispieldaten: sound.media.mit.edu/KEMAR.html

absolute Hörschwelle

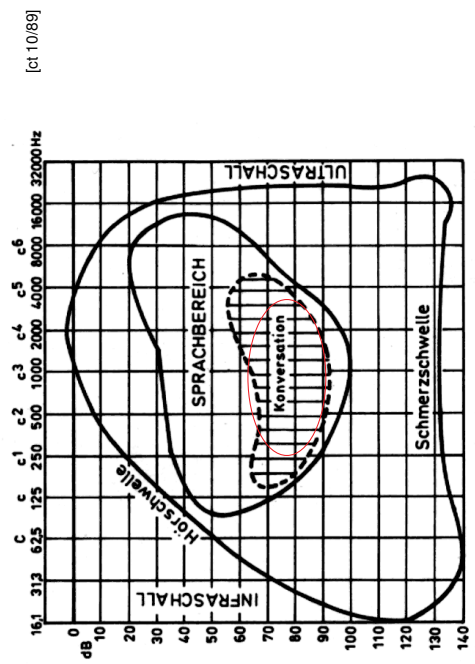


[Terhardt]

Abb. 9.11. Monaurale Absoluthörschwelle für Sinustöne. Glatte Kurve: Mittelwert des Schwellenpegels L_A von normalhörenden Personen. Gezackte Kurve: Mit der Methode des pendelnden Regelns (Békésy tracking) gemessen (25jähriger Mann, rechtes Ohr). Abszisse: Tonfrequenz, SPINC-skaliert; Ordinate: Schallpegel

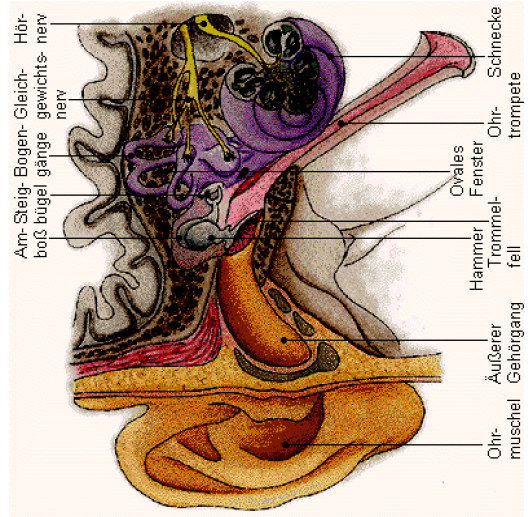
- geringe Empfindlichkeit bei tiefen und hohen Tönen
 - maximale Empfindlichkeit zwischen 2..5 KHz
- => Loudness-Kurven

Hörbereich und Sprachbereich



gute Sprachverständlichkeit im "Hauptsprachbereich" (ca. 300-3500 Hz)

Ohr: Schema



drei Untereinheiten:

- Außenohr
 - Mittelohr
 - Innenohr
- bzw.:
- Verstärkung, Delay
 - Impedanzwandlung
 - Frequenzanalyse

(glasklar Mensch 3D)

Gehör: Modell der Signalverarbeitung

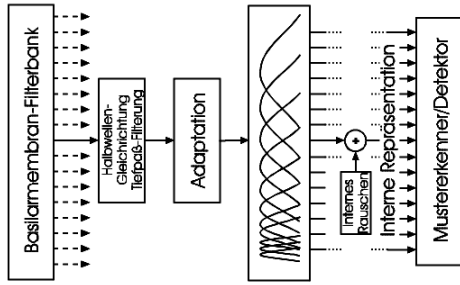
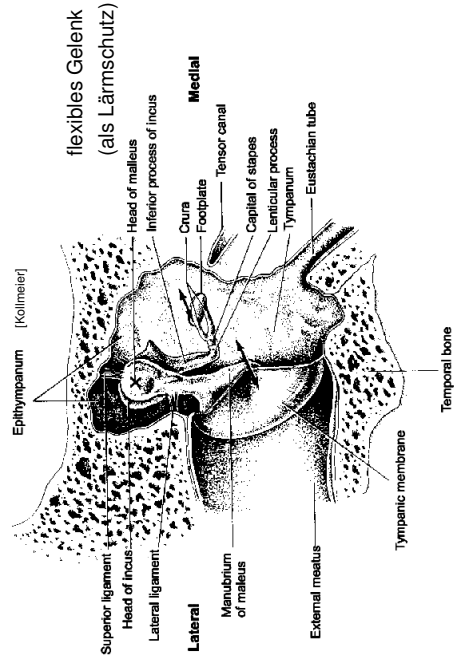


Abbildung 1.4: Modell der effektiven Signalverarbeitung nach Dau et al. (1987) [Kollmeier]

Digitale Audioverarbeitung | WS 2000 | 18.205

Mittelohr



- Verstärkung (Impedanzwandlung), Übergang Luft nach Wasser
- Trommelfeld ca. 20fache Fläche des ovalen Fensters

Digitale Audioverarbeitung | WS 2000 | 18.205

Ohr: Schema

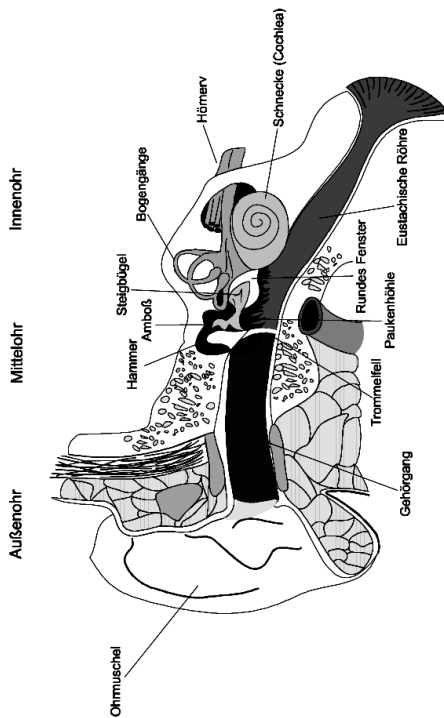


Abbildung 1.1: Übersicht über den Aufbau des Gehörs [Kollmeier]

Digitale Audioverarbeitung | WS 2000 | 18.205

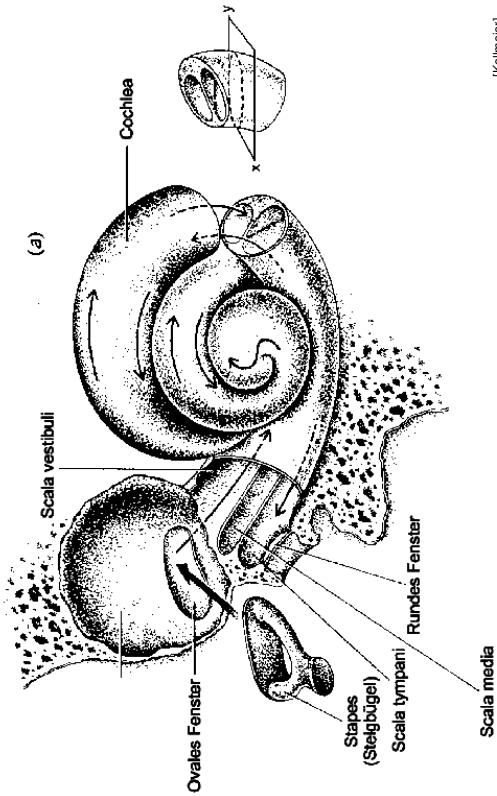
Gehör: Teile, Funktion, Störungen

Anteil des Gehörs	Funktion	Störung	Ätiologische Diagnosezeit
Außenohr	Richtungsabh., Filterung / Bündelung	Ohrmitbildung	klinisch
Mittelohr	Impedanztransformation Luft → Wasser	Schallleitungs-Schwachhörigkeit (z.B. Erguß, Otitis media)	Otoskopie Impedanzaudiometrie Stimmgabeltest Ton-Audiogramm Otoskopische Emissionen Überschwellige Tests
Innenohr	Umwandlung Wasserschall → Nerven- erregung, Frequenz-Orts-Transformation	Cochleäre Schwerhörigkeit (z.B. Lärm, Alter, ...)	Hirnstamm-Audiometrie Sprachaudiometrie Bildgebende Verfahren
Hörnerv / Hirnstamm	Kodierung akustischer Information, Auswertung interauraler Unterschiede, Modulations-Kodierung	Retrocochleäre (neurale) Schwerhörigkeit, Lokalisationsstörung	Zentrale Sprachtests
Cortex	Sprachwahrnehmung, komplexe Verar- beitung	zentrale Hörstörungen (z.B. Aphasie)	

[Kollmeier]

Digitale Audioverarbeitung | WS 2000 | 18.205

Innenohr: Aufbau und Cochlea



[Kollmeier]

Innenohr: cortisches Organ

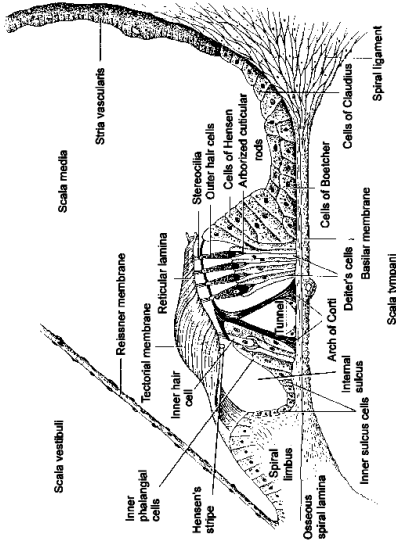
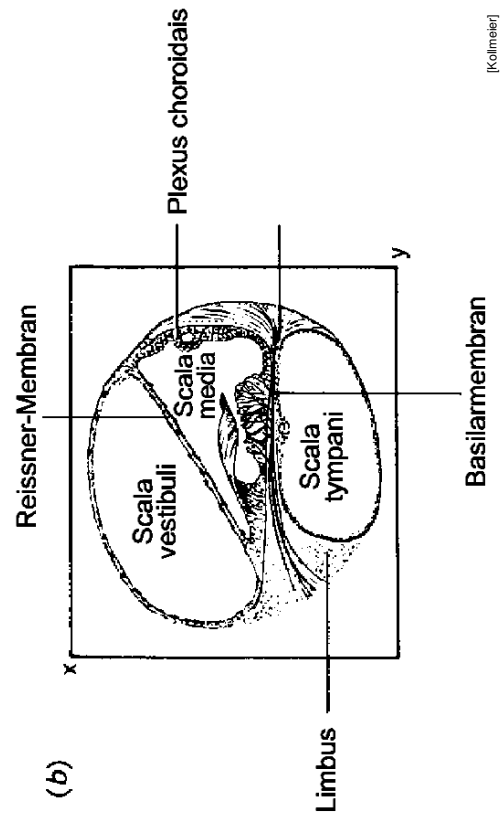


Abbildung 3.7: Schematische Darstellung des Cortischen Organs (aus Gulick, W. L., Gescheider, G. A., Frisina, R. D.: Hearing. Oxford University Press, Oxford 1989)

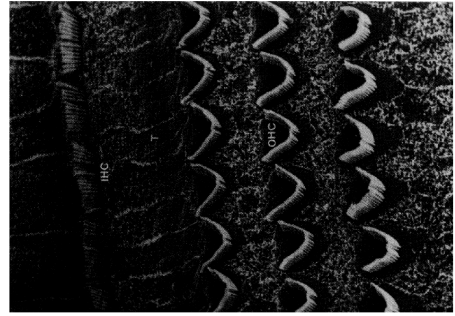
[Kollmeier]

Innenohr: cortisches Organ



[Kollmeier]

Innenohr: Haarzellen



innere Haarzellen
"Sensoren"

äußere Haarzellen
"Aktoren"

Abbildung 3.8: Detailansicht der Innenohr-Anatomie, IHC = Innere Haarzellen, OHC = Äußere Haarzellen (aus Dallos, P., Popper, A. N., Fay, R. R. (Hrsg.): The Cochlea, Springer-Verlag, New York 1986)

Cochlea und Nerven

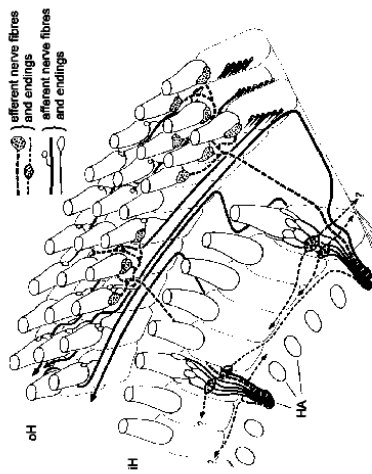


Abbildung 4.5: Schematische Anknüpfung der Haarzellen an den Hörnerven (oH: äußere Haarzellen, iH: innere Haarzellen) (aus Luce, R.D.: Sound and Hearing. Lawrence Erlbaum Verlag, New Jersey 1993) [Kollmeier]

Basilarmembran

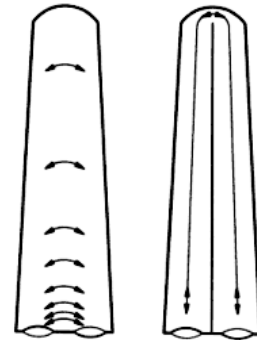


Abb. 9.9. Wenn die Basilarmembran nicht vorhanden wäre (obere Skizze, ausgestreckter Cochleakanal), würde die Flüssigkeitsbewegung sich bei Auslenkung des ovalen Fensters wegen der inneren Reibung auf den basalen Teil (links) konzentrieren. Wenn dagegen der Kanal über einen Teil seiner Länge durch eine elastische Membran unterteilt wird (unten), wird bei sehr langsamer Auslenkung die Strömung um die Membran herum gezwungen, während rasche Auslenkung wegen der Massenträgheit der bewegten Flüssigkeitssäule zur Querauslenkung der Membran führt, und zwar umso weiter links, je rascher die Auslenkung erfolgt [Terhardt]

- mechanische Lösung zur Frequenzanalyse

Cochlea: Halbwellengleichrichtung

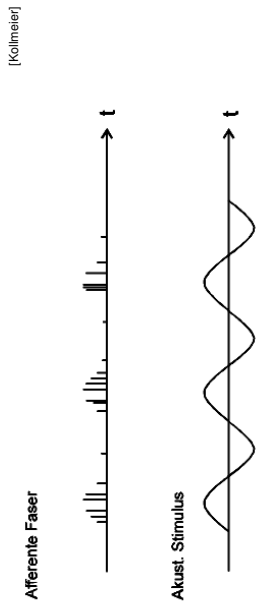


Abbildung 4.6: Halbwellengleichrichtung bei der Kodierung eines akustischen Stimulus in der Haarzelle bzw. im nachgeschalteten auditorischen Nerv (schematisch). Dargestellt ist die Aktivität des Hörnerven, die nur während der positiven Halbwellen des Eingangsstimulus auftritt [Kollmeier]

- innere Haarzellen reagieren nur auf Dehnung
- Nerven-Spikes nur für positive Halbwellen

Haarzellen: otoakustische Emission

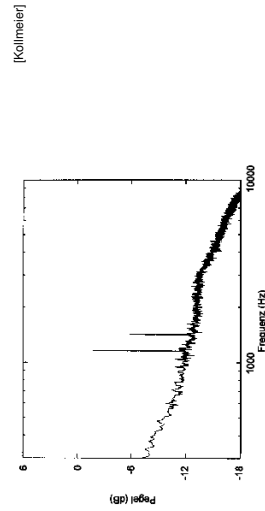


Abbildung 3.17: Gemitteltetes Leistungsspektrum bei der Bestimmung einer spontanen otoakustischen Emission, die durch vereinzelte Spitzen im Spektrum gekennzeichnet ist. (aus Uppenkamp, S.: Grundlagen und Anwendung der otoakustischen Emissionen zur objektiven Hörprüfung. Dissertation, Universität Göttingen 1992) [Kollmeier]

- Haarzellen sind aktive Steilglieder
- harmlose spontane Emission, bis zum Tinnitus

Lautheit

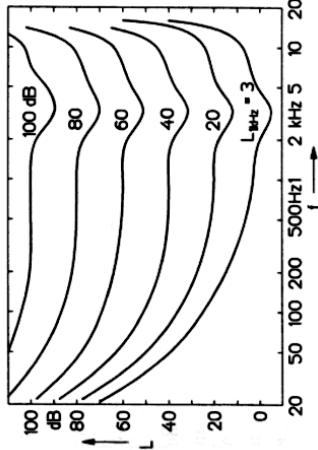


Abb. 10.8. Kurven gleicher Lautheit von Sinustönen im ebenen Schallfeld: Schallpegel L eines Sinustones (Ordinate), welcher die gleiche Lautheit hervorruft wie ein 1 kHz-Ton mit dem angegebenen Pegel, in Abhängigkeit von der Tonfrequenz f (Abszisse)

- bei leisen Tönen schwache Wahrnehmung von Höhen/Bässen
- Empfindlichkeitsmaximum bei ca. 3 KHz
- "loudness"-Regler am HiFi-Verstärker zur Kompensation

Tuningkurven: Schema

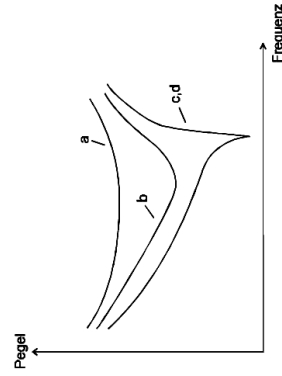


Abbildung 3.6: Schematischer Verlauf von „Tuning-Kurven“, d. h. des Pegels eines Sinustons, der zum Erreichen einer bestimmten Auslenkung der Basilar membran (bzw. zu einer bestimmten Erhöhung der Nervenfaser-Feuerrate gegenüber der Spontarrate) notwendig ist, als Funktion der Frequenz. Dargestellt sind schematisch die Ergebnisse für (a) optisch/stroboskopisch (v. Békésy), (b) Mößbauer-technik, (c) neuere Laser-Interferometrische Messungen, (d) neurophysiologisch

Ohr und Hörbahn

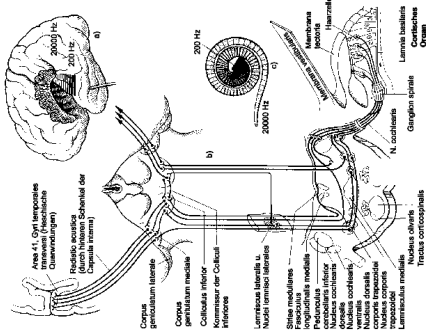


Abbildung 4.12: Übersicht über die Anatomie der Hörbahn (aus Duus, P.: Neurologisch-topische Diagnostik, Thieme Verlag, Stuttgart 1995). [Kollmeier]

Inter-Spike Histogramme

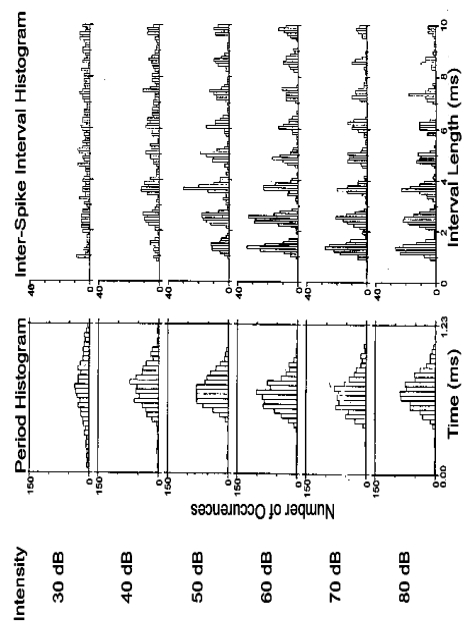
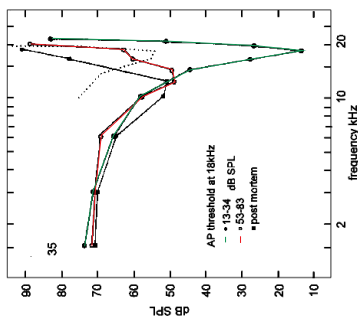


Abbildung 4.9: Periodenhistogramm und Inter-Spike-Interval-Histogramm für eine Hörnervefaser bei systematischer Erhöhung der Intensität eines sinusförmigen Reizes von ca. 800 Hz von 30 dB auf 80 dB. [Kollmeier]

Tuningkurven: gemessen



- hohe Güte
- keineswegs passiv
- aktive Regelung

Abbildung 4.7: Schematische Tuning-Kurve (Reizintensität an der Schwelle als Funktion der Frequenz des Stimulus) für zwei verschiedene sensitive Hörmervenfasern sowie für den Zustand nach abgestellter Blutzufuhr (post mortem). (aus Moore, B. C. J.: Frequency Selectivity in Hearing. Academic Press, London 1986)

[Kallmeier]

Maskierung

Mithörschwelle eines Sinustones bei Verdeckung durch Schmalbandrauschen. In Abb.11.4 sind drei Mithörschwellen von Sinustönen verdeckt durch Schmalbandrauschen der angegebenen Mittenfrequenzen dargestellt [1125]. Die Fourier-Spektren der maskierenden Schmalbandgeräusche sind durch Verwendung von Bandpassfiltern hoher Flankensteilheit auf Bandbreiten von ungefähr 100, 160 und 700 Hz beschränkt [CD 2].

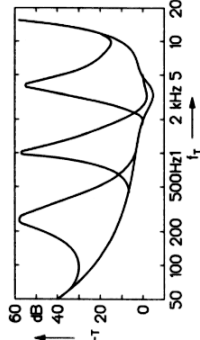
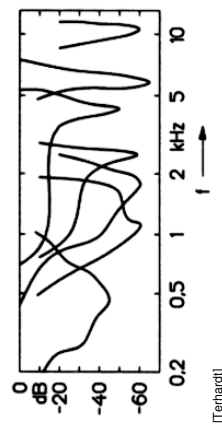


Abb. 11.4. Mithörschwellen von Sinustönen bei Verdeckung durch frequenzgruppenbreite Schmalbandrauschen mit den Mittenfrequenzen 250, 1000, 4000 Hz und dem Maskierpegel $L_M = 60$ dB. Nach [1125]

[Terhardt]

- Frequenzanalyse in der Cochlea nicht perfekt
- laute Töne verdecken benachbarte leisere
- Ausnutzung in psychoakustischen Modellen (MiniDisc, MP3, usw)

Tuningkurven: Katzen

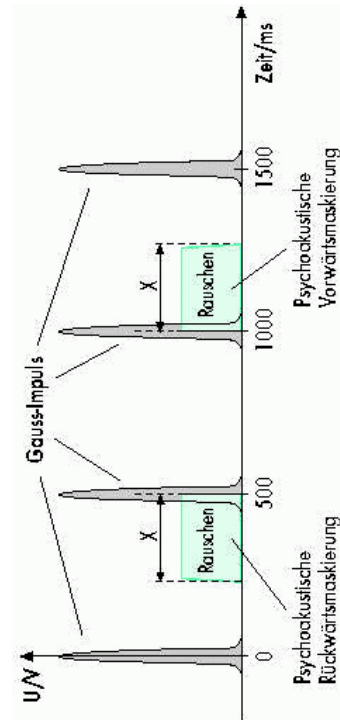


[Terhardt]

Abb. 9.10. Tuningkurven, gemessen an anästhesierten Katzen. Jede Kurve gehört zu einer anderen Faser des akustischen Nerven. Die Ordinate gibt denjenigen Schallpegel eines Sinustones an, welcher zur Erregung einer festen Anzahl von Aktionspotentialen pro Zeiteinheit erforderlich ist. Abszisse: Frequenz des Sinustones. Nach King [50]

- hohe Güte der Frequenzselektion
- "aktive" Frequenzbereiche der Fasern überlappen
- menschliches Gehör entsprechend

Maskierung: temporal



[cf]

- laute Töne verdecken spätere leisere Töne
- und frühere leisere Töne (!)
- Ausnutzung in psychoakustischen Modellen (MiniDisc, MP3, usw)

Frequenzauflösung

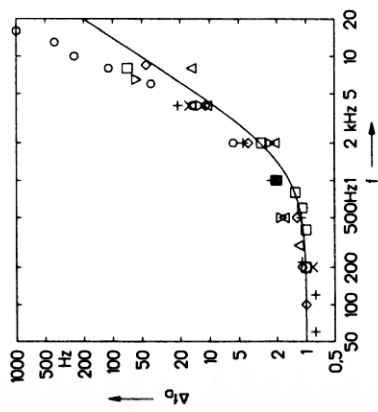


Abb. 9.19. Messergebnisse zum eben wahrnehmbaren Frequenzunterschied zwischen aufeinanderfolgenden Sinustönen (Ordinate, Δf_0) in Abhängigkeit von der Tonfrequenz f (Abszisse).
 Kreise: Henning [434]; Quadrate: Wier et al. [1064]; aufwärtsgerichtete Dreiecke: Fastl [280]; abwärtsgerichtete Dreiecke: Moore & Glasberg [645]; Rauten: Walliser [1040]; x: Rosenblith & Stevens [795]; +: Harris [412]. Die durchgezogene Kurve gibt die angenäherte Repräsentation der Messergebnisse durch (9.70) an [Tenhardt]

- bis 1 Hz Frequenzauflösung (bei geübten Personen)

Tuningkurven vs. Tonhöhe

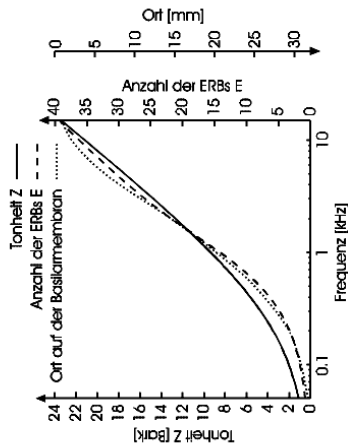


Abbildung 5.3: Mel-Skala, Bark-Skala, ERB-Skala

- wahrgenommene Tonhöhe korrespondiert mit Position
- passende Skalen

Frequenzgruppen

Tabelle 9.1: Frequenzgruppen nach Zwicker 1982

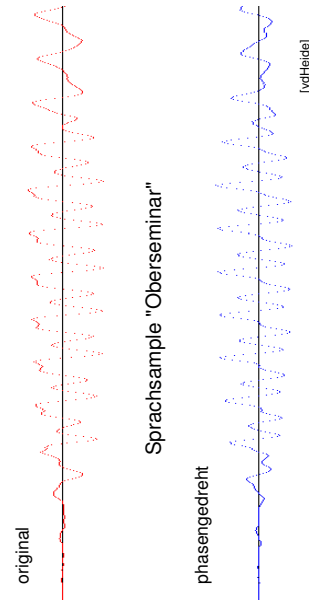
z/Bark	f_z /Hz	f_0 /Hz	Δf_z /Hz	f_m /Hz
0	100	0	100	50
1	200	100	100	150
2	300	200	100	250
3	400	300	100	350
4	510	400	110	450
5	630	510	120	570
6	770	630	140	700
7	920	770	150	840
8	1080	920	160	1000
9	1270	1080	190	1170
10	1480	1270	210	1370
11	1720	1480	240	1600
12	1790	1720	280	1850
13	2000	2000	320	2150
14	2320	2320	380	2500
15	2700	2700	450	2900
16	3150	3150	550	3400
17	3700	3700	700	4000
18	4400	4400	900	4800
19	5300	5300	1100	5800
20	6400	6400	1300	7000
21	7700	7700	1800	8500
22	9500	9500	2500	10500
23	12000	12000	3500	13500
24	15500	15500		

[Zwicker]

Tonhöhenwahrnehmung?

- Messungen
- vgl. Tuningkurven
- tiefe Frequenzen: ca. 100 Hz Bandbreite
- hohe Frequenzen: log. Breite der Gruppen => "Bark" Frequenzskala
- Ausgangspunkt für psychoakustische Modelle

Phasen: Hilbertfilter



- Ohr unempfindlich gegen Phaseninformation
- verändertes Signal klingt einzeln (vollkommen) gleich
- erst bei Interferenzen mehrerer Signale hörbar
- Phasengang vieler Systeme (Telefon!) extrem schlecht

Klangfarben

akustisches Signal mit vielen Teiltönen:

- keine Wahrnehmung aller Einzelöne
- sondern Zuordnung zu Quellen/Instrumenten
- tiefste Frequenz bestimmt die wahrgenommene Tonhöhe
- Obertonspektrum identifiziert die Quelle
- mit Eigenschaft "Klangfarbe"
- Klangfarbe auch stark vom "attack" des Klangs abhängig (=> Synthesizer mit kurzen Samples plus Schleifen)
- entsprechend automatische "Sprecheridentifikation"
- komplexe Zusammenhänge für Akkordwahrnehmung

Klangfarben: Spektrogramm



Abb. 1.9. Beispiel eines Teiltonzeitmusters. Ausschnitt einer mehrstimmigen Blasmusikdarbietung. Die hauptsächlich beteiligten Instrumente sind Trompete, Posaune, Klarinette und Tuba. Die instantane Teilton-Amplitude wird durch die Strichstärke angegeben. Ordinate Bark-skaliert. Nach [969]

[Terhardt]

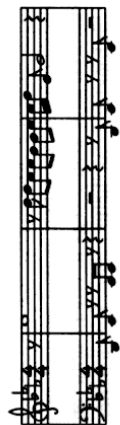


Abb. 1.10. Transkription der Trompeten- und der Tuba-Stimme des Musikbeispiels, dessen Teiltonzeitmuster in Abb. 1.9 dargestellt ist

Klangfarben: Beispiele

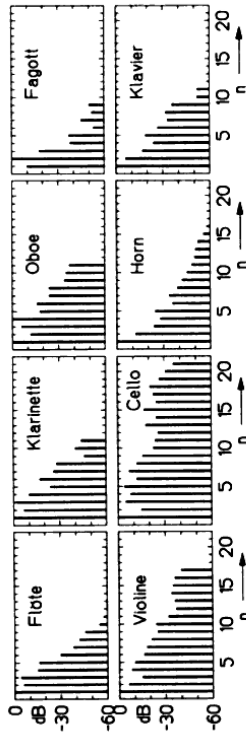
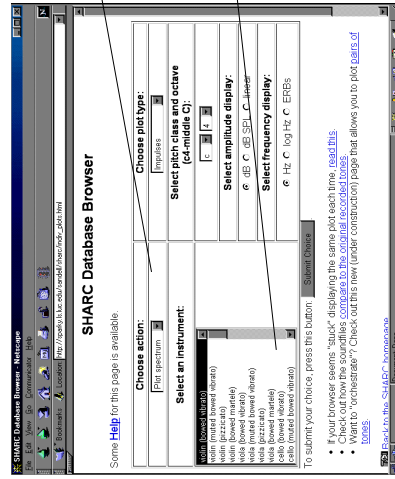


Abb. 8.12. Teiltonspektren verschiedener Musikinstrumente schematisch

[Terhardt]

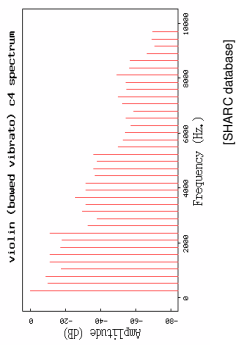
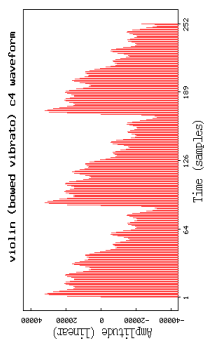
- Streicher mit reichem Obertonspektrum
- Flöten mit sehr reinen Tönen (insb. Blockflöte)
- Klavier stark anschlag- und tonhöhenabhängig
- aber auch "Attack-Phase" maßgeblich für Klangwahrnehmung

Klangfarben: SHARC-Datenbank



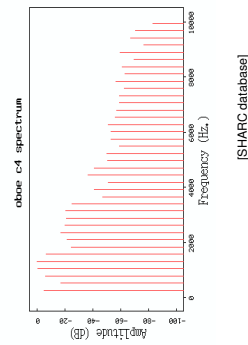
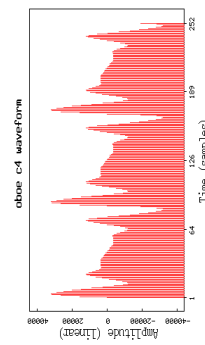
- www.sparky.is.luc.edu/sandell/sharc/indiv_plots.html
- schönes Web-Interface zu Samples und Spektren
- kein Zusammenhang mit den "SHARC" Signalprozessoren

Klangfarben: Violine



- Sägezahnform: reiches Obertonspektrum
- auch viele gerade (angenehme) Obertöne
- typisch für Streichinstrumente

Klangfarben: Oboe



- irreguläre Wellenform
- außergewöhnlich reiches Obertonspektrum
- andere Blasinstrumente mit weit weniger Obertönen
- z. B. Blockflöten fast reine Sinusklänge (plus Anblasgeräusche)

Sprache

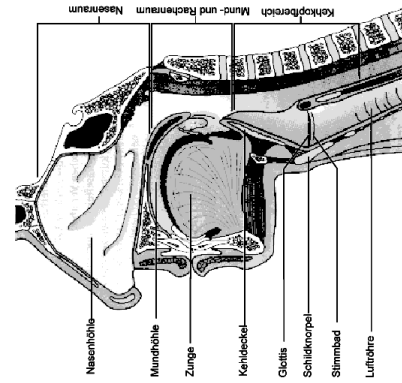
"niedere" Aspekte:

- Sprachsignale, Spektren, Formanten, ...
- Trennung stimmhafte / stimmlose Laute
- Synthese / Analyse von Sprachsignalen
- low-bitrate Kodierung (z.B. für Mobiltelefone)

"höhere" Sprachverarbeitung:

- NATS (von Hahn, Menzel)
- WSV (Habel, Freksa)

Sprache: Vokaltrakt



- Glottis, Stimmband
- Klang zusätzlich durch Zungen-/Lippenstellung
- typische Spektren: Mann / Frau / Kind normal / erregt

Abbildung 6.1: Vokaltrakt im Querschnitt (schematisch) (aus Kießling, J., Kollmeier, B., Diller, G.: Versorgung und Rehabilitation mit Hörgeräten. Thieme Verlag, Stuttgart 1997) [Kollmeier]

Sprache: Spektrogramm "Durst"

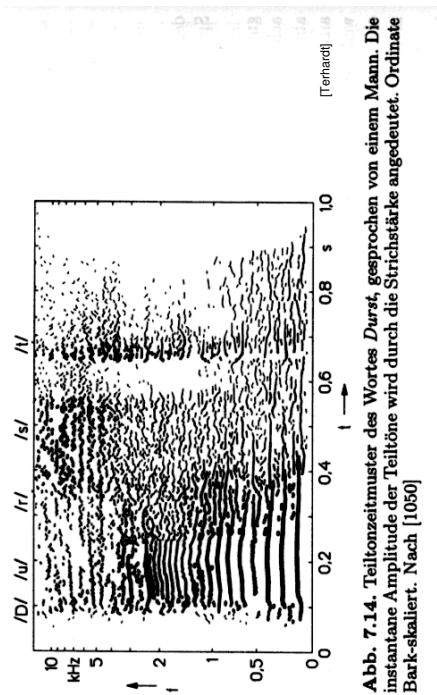


Abb. 7.14. Teilzeitmuster des Wortes *Durst*, gesprochen von einem Mann. Die instantane Amplitude der Teiltöne wird durch die Strichstärke angedeutet. Ordinate Bark-skaliert. Nach [1050]

- außerordentlich komplex: Problem Spracherkennung

Sprache: Spektrogramm

Formanten, d. h. die Frequenzbereiche, bei denen der Vokaltrakt eine besonders hohe Verstärkung des akustischen Signals bewirkt. Die Formanten stellen sich damit als Spitzen im Spektrum dar und sind charakteristisch für den jeweils artikulierte Vokal (s. unten).

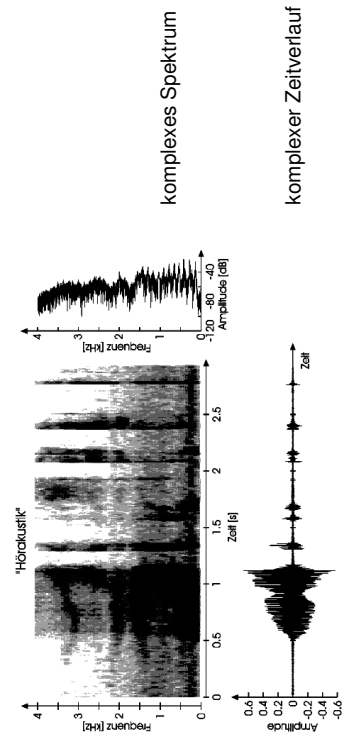


Abbildung 6.2: Zeit-Signal (unten), Spektrum (rechts) und Spektrogramm (links oben) des Wortes „Hörakustik“ [Kollmeier]

Sprache: Spektren, Formanten

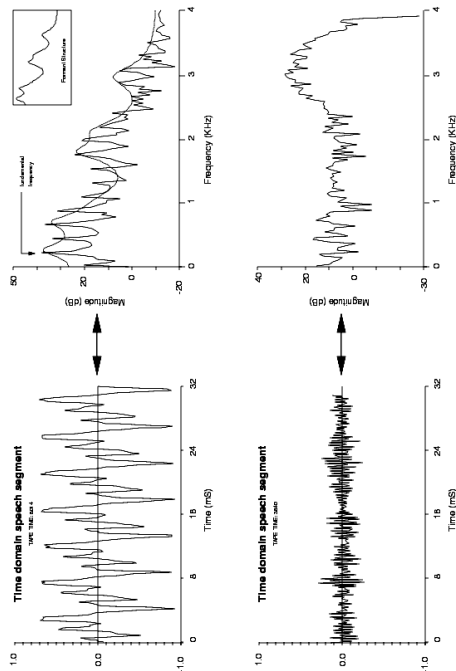


Fig. 1. Voiced and Unvoiced Segments and their short-time spectra. [Spanias]

Sprache: Artikulation

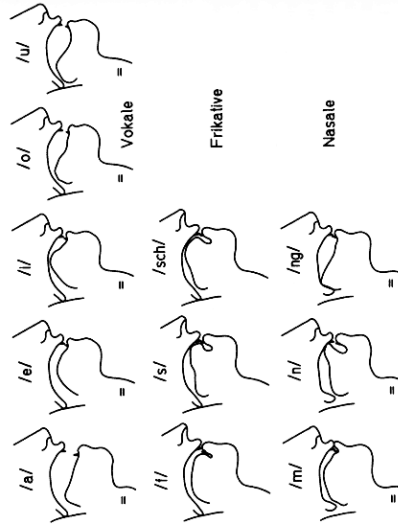


Abb. 7.1. Schematische Darstellung der Artikulation einiger Sprachlaute

- Zungen- und Lippenstellung maßgeblich

[Terhardt]

Sprache: Vokaltrakt

• **Stimmband + Resonanzrohr**
 • **Legen der Formanten:**
 Wellenlänge: $L/4 = 17\text{cm}$
 $F1 = c/L = 340\text{m/s} / 68\text{cm} = 500\text{ Hz}$
 $F2 = 1500\text{ Hz}$
 $F3 = 2500\text{ Hz}$
 $F3 = 3500\text{ Hz}$

Sprache: Teiltonspektren der Vokale

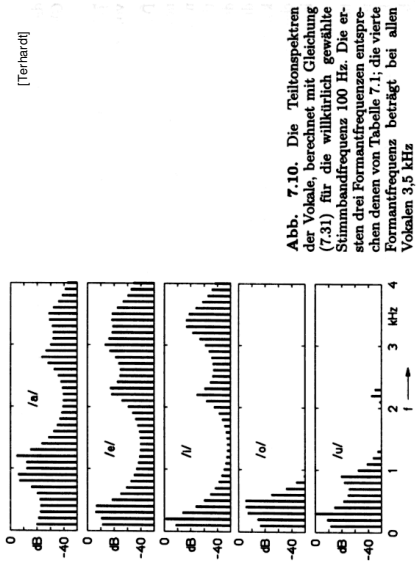


Abb. 7.10. Die Teiltonspektren der Vokale, berechnet mit Gleichung (7.31) für die willkürlich gewählte Stimmbandfrequenz 100 Hz. Die ersten drei Formantfrequenzen entsprechen denen von Tabelle 7.1; die vierte Formantfrequenz beträgt bei allen Vokalen 3,5 kHz

- Spektren der Vokale, Formanten deutlich sichtbar

Sprache: Konsonanten bdg

• **Stimmband + Resonanzrohr**
 • **Legen der Formanten:**
 Wellenlänge: $L/4 = 17\text{cm}$
 $F1 = c/L = 340\text{m/s} / 68\text{cm} = 500\text{ Hz}$
 $F2 = 1500\text{ Hz}$
 $F3 = 2500\text{ Hz}$
 $F3 = 3500\text{ Hz}$

Sprache: Vokaldreieck

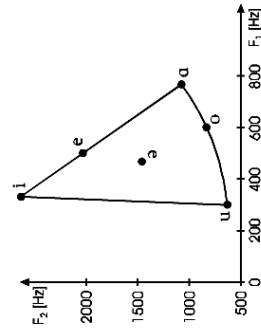


Abbildung 6.1.4: Vokaldreieck. Aufgetragen ist die Frequenz des ersten Formanten F_1 (Abszisse) und des zweiten Formanten F_2 (Ordinate) für unterschiedliche Vokale (schematisch) [Kollmeier]

- typische Frequenzen / Formanten der Vokale
- verhältnismäßig einfache Erkennung von stimmhaften Lauten

Lärm: Pegelbereiche

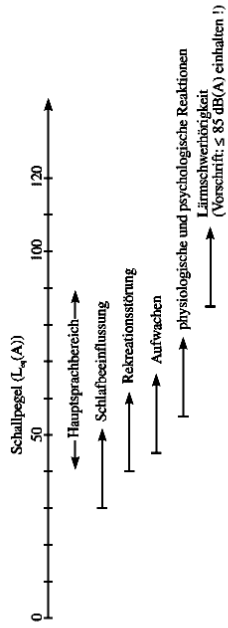
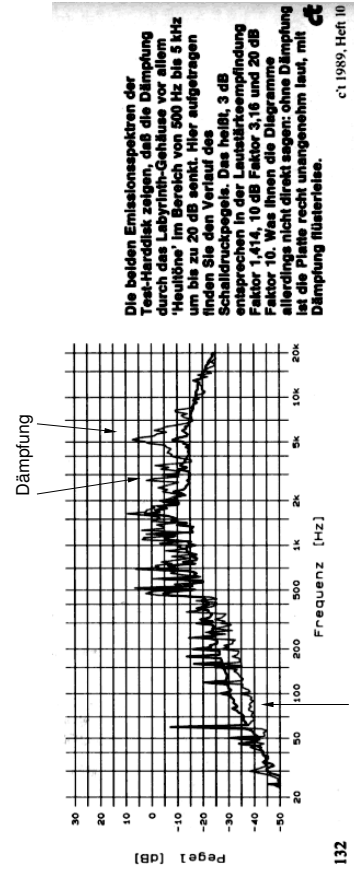


Abbildung 3.12: Pegelbereiche, für die unterschiedliche Schallwirkungen beim Menschen auftreten [Kollmeier]

- Lärmpegel an Straßen selten unter 50 dB(A) ...
- laufender Computer:
- Nadeldrucker: 70+ dB(A)

Lärm: Festplatte normal und gedämpft



Die beiden Emissionsspektren der Test-Harddisk zeigen, daß die Dämpfung durch das Laubhörn-Gehäuse vor allem 'Heulöne' im Bereich von 500 Hz bis 5 kHz um bis zu 20 dB senkt. Hier aufgetragen finden Sie den Verlauf des Schalldruckspekts. Das heißt, 3 dB entsprechen in der Lautstärkeempfindung Faktor 1,414, 10 dB Faktor 3,16 und 20 dB Faktor 10. Was Ihnen die Diagramme allerdings nicht direkt sagen: ohne Dämpfung ist die Platte recht unangenehm laut, mit Dämpfung flüsterleise.

c1 1989, Heft 10

hier sogar lauter

- Schalldämpfung ist nicht einfach...

Sprache: Vokalkategorien

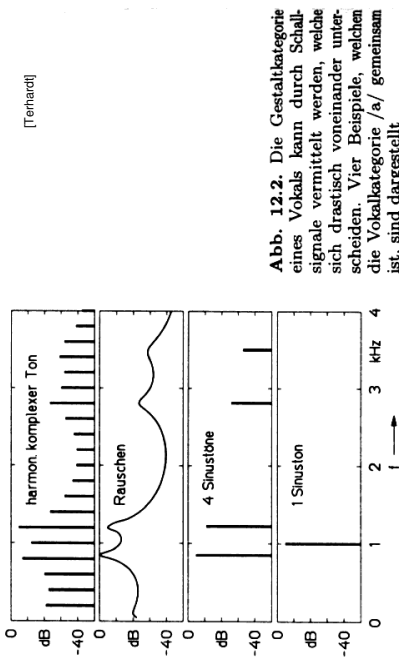
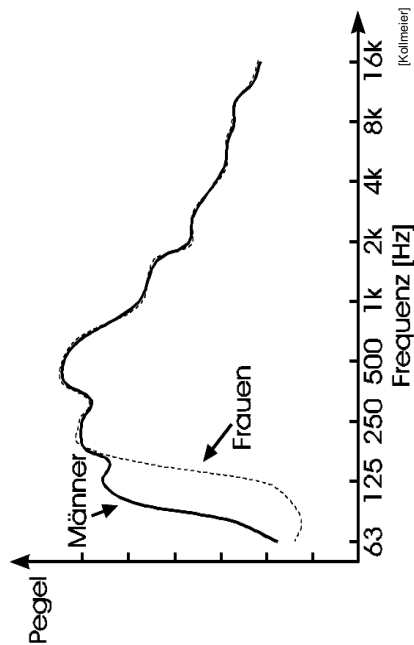


Abb. 12.2. Die Gestaltkategorie eines Vokals kann durch Schallsignale vermittelt werden, welche sich drastisch voneinander unterscheiden. Vier Beispiele, welchen die Vokalkategorie /a/ gemeinsam ist, sind dargestellt

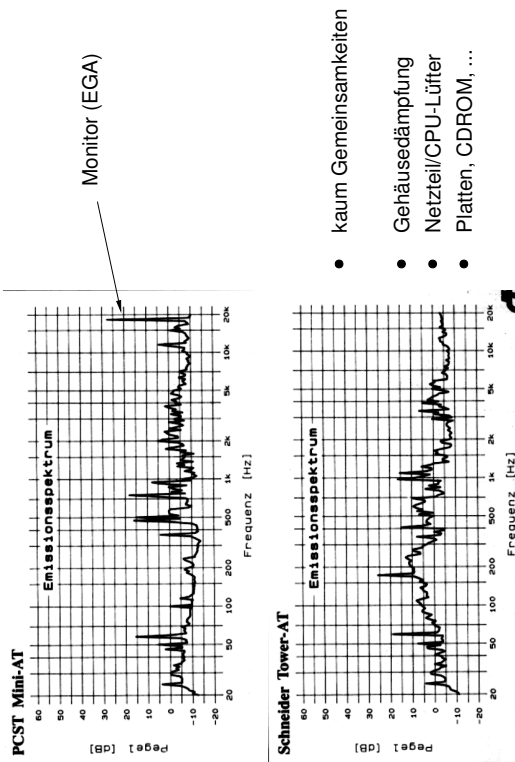
- Wahrnehmung: jeweils als Vokal /a/ => perceptual speech coding

Sprache: Frequenzbereiche



- Frequenzspektrum oberhalb 200 Hz identisch
- Männer mit tieferen Frequenzen zw. 60 und 250 Hz

Lärm: Emissionsspektren, 1989



Digitale Audioverarbeitung | WS 2000 | 18.205

Lärm: Schädigungen

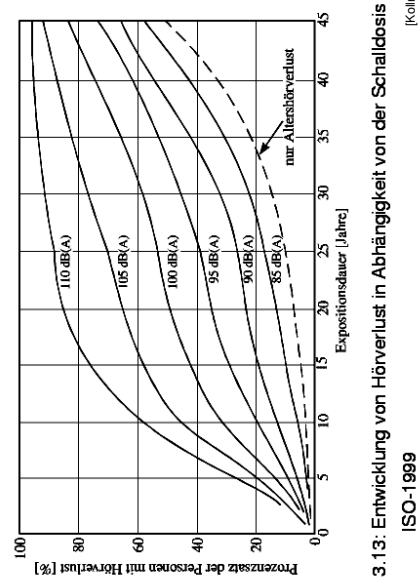
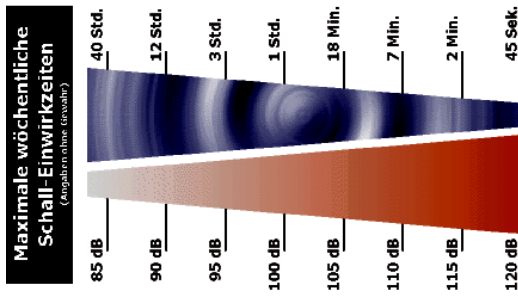


Abbildung 3.13: Entwicklung von Hörverlust in Abhängigkeit von der Schalldosis gemäß ISO-1999

- bei höheren Pegeln Schäden innerhalb von Sekunden

Digitale Audioverarbeitung | WS 2000 | 18.205

Lärm: Schädigung



Digitale Audioverarbeitung | WS 2000 | 18.205

Gehör: Zusammenfassung

Gehör außerordentlich leistungsfähig:

- überdeckt 13 Größenordnungen Schalldruck
 - fast optimale Empfindlichkeit
 - Hörbereich etwa 20Hz .. 20 KHz
 - Frequenzanalyse "mechanisch" in der Cochlea
 - komplexe Verarbeitung/Erkennung im Gehirn
 - Wahrnehmung Klang vs. Geräusch
- insbesondere zur Raumwahrnehmung:
- Zeit- und Intensitätsdifferenzen der Signale
 - Kopfübertragungsfunktion
 - Auswertung von Echos

=> entsprechende Modelle für effiziente Signalverarbeitung

Digitale Audioverarbeitung | WS 2000 | 18.205

Digitale Signalverarbeitung

- Motivation
- analoge vs. digitale Signalverarbeitung
- Literatur
- Zahlenfolgen, LTI-Systeme
- Abtasttheorem
- Spektrum, FFT, Übertragungsfunktion
- Quantisierung, AD/DA Konverter, Dithering
- Audio-Algorithmen
- digitale Filter
- Dynamikbeeinflussung
- Raumsimulation, Hall

DSP: Literatur

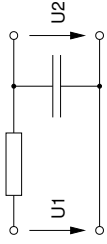
- K.v.d.Heide, Skripte zu "Signalverarbeitung" und "Nachrichtentechnik", FB Informatik, Uni HH, tech-www.informatik.uni-hamburg.de/lehre/ (Matlab)
- Mathworks, Inc., Matlab 5.3 User and Toolbox Manuals, www.matlab.com
- U. Zölzer, Digitale Audiosignalverarbeitung, Teubner 1996
- P. Gerdßen, P. Kröger, Digitale Signalverarbeitung in der Nachrichtenübertragung, Springer 1997
- R. W. Hamming, Digital Filters, Prentice Hall, 1983
- W.H.Press, B.P.Flannery, S.A.Teukolsky, W.T. Vetterling, Numerical Recipes, Cambridge Univ. Press
- U. Tietze, Ch. Schenk, Halbleiter-Schaltungstechnik, Springer 1983 (analoge Schaltungen)

- IEEE Journal Signal Processing
- diverse Konferenzen zum Thema, Audio u.a. Audio Engineering Convention
- DSP-Datenbücher (www.motorola.com, www.analog.com)

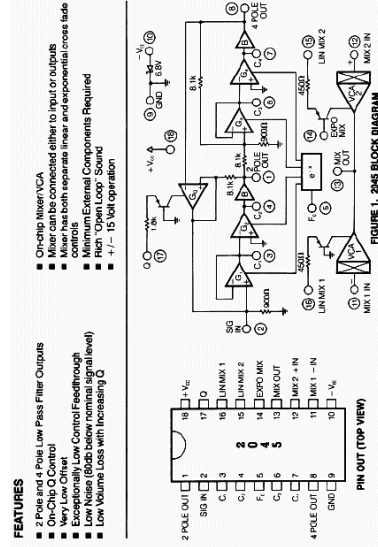
DSP: Motivation

analoge Signalverarbeitung:

- Modellierung mit Differentialgleichungen
 - direkte Umsetzung mit diskreten Bauteilen
 - hauptsächlich mit "analoger" Elektronik
 - Widerstände, Kondensatoren, Spulen, Transistoren
- aber:
- entweder geringe Genauigkeit (z.B. 10%) oder sehr teuer
 - Alterung, Exemplarstreuungen, aufwendiger Abgleich
 - wirtschaftlich nur für geringe Anzahl der Bauelemente
 - Datenspeicherung (analog) problematisch
- => komplexe Algorithmen nicht umsetzbar
- => digitale Verarbeitung verspricht Abhilfe

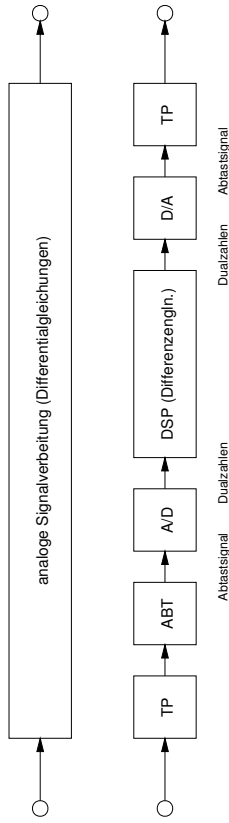


Beispiel für eine Audio-Analogschaltung



- SSM-2045: Analogfilter mit Opamps
- typische Komplexität: einige Dutzend Komponenten
- Probleme mit Rauschen und Stabilität

DSP: vs. analoge Verarbeitung



"analoge" Signalverarbeitung "digitale"

- | | |
|---|--|
| <ul style="list-style-type: none"> • zeitkontinuierlich • wertkontinuierlich • Differentialgleichungen • Drift, Rauschen, ... | <ul style="list-style-type: none"> • zeitdiskret (Abtastung) • wertdiskret (Quantisierung) • Differenzgleichungen • Überlauf, Rundungsfehler |
|---|--|

DSP: Audio-Anwendungen

Signalverarbeitung im Audio-Bereich:

- Datenspeicherung (CD, DVD, ...)
- Datenkompression (MP3, AC3, MiniDisc, ...)
- Datenübertragung (DSR, GSM, ...)
- jeweils für Musik und Sprache
- Musikproduktion (digitales Tonstudio)
- Musiksynthese (z.B. virtuell analog)
- Audioeffekte (z.B. Hall)
- Surround-Verfahren (z.B. DirectX mit HRTF)
- Samplefrequenz typ. 44.1 / 48 KHz
- hohe Genauigkeit erforderlich: 16 .. 24 bit

DSP: Definition

"digital signal processing" (digitale Signalverarbeitung):

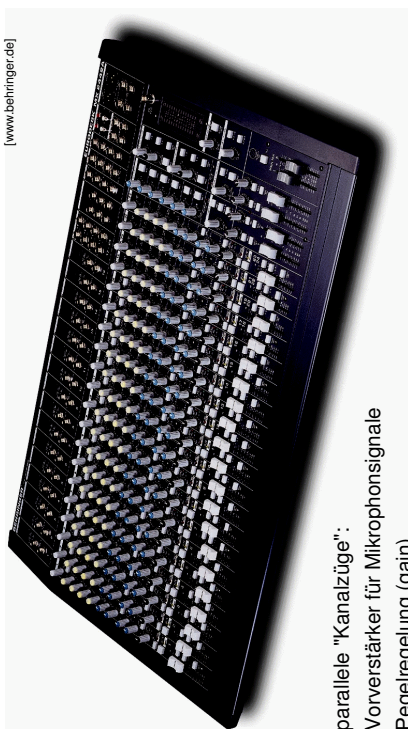
- Verbeitung von
- zeitdiskreten
 - wertdiskreten
 - Zahlenfolgen
- auf Digitalrechnern
- für:
- Nachrichtenübertragung
 - Bildverarbeitung
 - Audioverarbeitung
 - usw.
- als Ersatz analoger Verfahren
 - Differenzen- statt Differentialgleichungen
 - oft mit Echtzeitanforderungen
 - im Prinzip beliebige Genauigkeit
 - oft mit Spezialhardware, etwa Signalprozessoren / ASICs

DSP: Eigenschaften

Verarbeitung zeit- und wertdiskreter Zahlenfolgen:

- als Ersatz für zeit/wert-kontinuierliche Verarbeitung (genügend hohe Abtastrate)
- wenn Abtasttheorem erfüllt (Wortbreite anpassen)
- beliebige Genauigkeit (z.B. keine Temperaturabhängigkeit)
- optimale Stabilität (z.B. Nullpunktseinstellung)
- kein Abgleich notwendig
- keine Alterung, keine Toleranzen
- auch für sehr komplexe Algorithmen
- die analog nicht (wirtschaftlich) realisierbar sind
- insbesondere: Datenspeicherung
- VLSI-Technologie erlaubt sehr hohe Performance

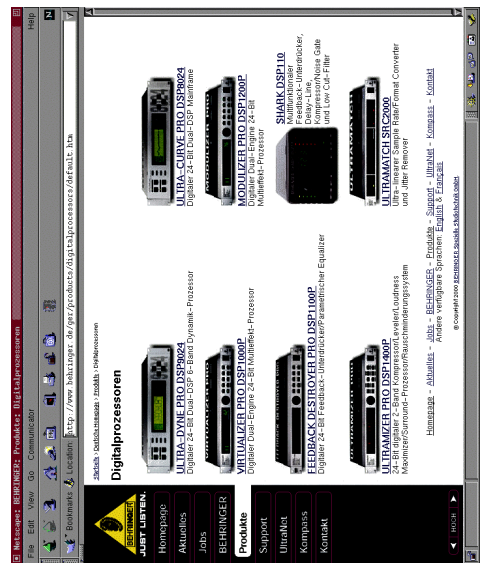
DSP: Beispiel Mischpult



[www.behringer.de]

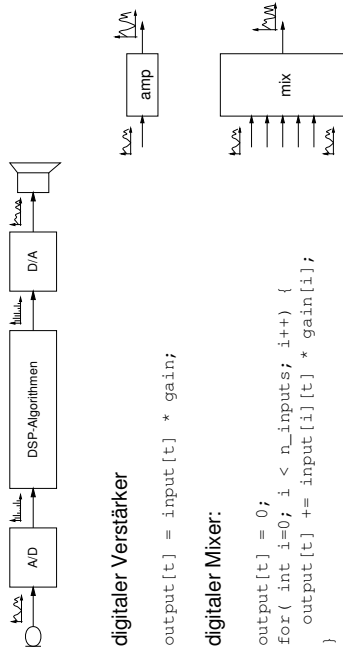
- parallele "Kanalzüge": Vorverstärker für Mikrophonsignale, Pegelregelung (gain), Klangregler und Dynamikregler
- Mischung der Einzelsignale und Routing auf "Busse"
- Mischung der Bussignale

DSP: Auswahl Digitalgeräte



Spektrum: Filter, Effekte, Kompressor, Feedbackkiller, ...

DSP: Verstärker, Mixer



- digitaler Verstärker
 $output[t] = input[t] * gain;$
- digitaler Mixer:

```
output[t] = 0;
for ( int i=0; i < n_inputs; i++) {
    output[t] += input[i][t] * gain[i];
}
```
- viele MAC-Operationen (multiply-accumulate)
- Problem Aussteuerung vs. Überlauf => saturation arithmetic

DSP: Echo, Hall, und mehr



- Verzögerung: auf alte Abtastwerte zurückgreifen
 $delay[t] = input[t]*gain + input[t-t2]*gain2;$
- Rekursion möglich: Echo, Hall, Raumsimulation
 $hall[t] = input[t]*gain + hall[t-t2]*gain2;$
- Algorithmen für viele Anwendungen: Verzögerung, Echo, Hall, Overdrive, Verzerrung, ... Filter, Formantfilter, Tonhöhenänderung, Tempoänderung, ...

DSP: Sampling und Synthese

- Sampling: Abtastwerte abspeichern

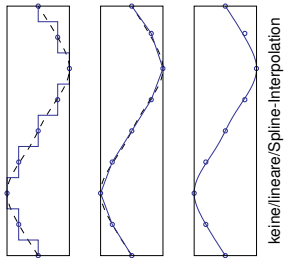
```
sample[t] = input [t];
```

- Samples direkt abspielbar (CD, Spiele)

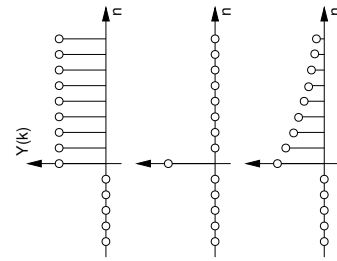
- "Wavetable"-Synthesizer:

```
output[t] = sample[t*pitch]
interpolate( sample[] )
filter( interpolate( sample[] ) )
effects( filter( interpolate( sample[] ) ) )
```

- Interpolation: nearest / linear / splines / ...



Zahlenfolgen, Elementarfolgen



Wertemenge {Y(k)}
Zuordnung: Index k -> Y(k)

wichtige "Elementarfolgen":

- Impulsfolge
- Sprungfolge
- Exponentialfolge
- Sinusfolge

lineare Systeme:

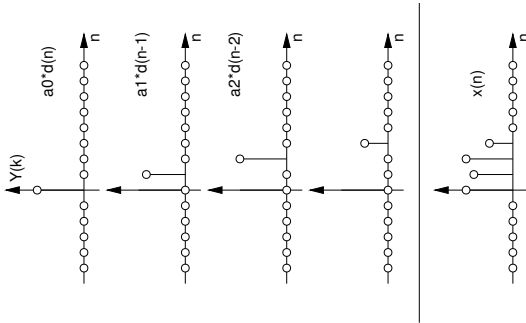
- Signale aus Elementarfolgen zusammensetzen
- Systemverhalten entsprechend berechenbar

Impulsfolge:

Konstruktion beliebiger Zahlenfolgen:

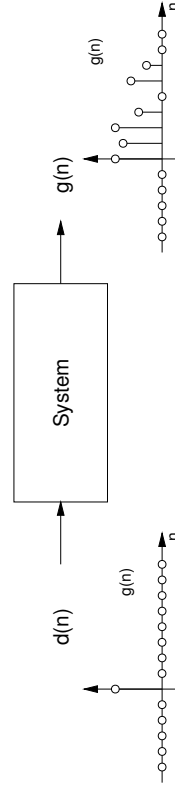
- Summe
 - skaliert
 - zeitverschoben
- Impulsfolgen
- $$x(n) = a_0 * d(n) + a_1 * d(n-1) + \dots$$

- gleichwertig: Konstruktion mit
- skalierten, zeitverschobenen
 - Sprungfolgen



Impulsantwort

"Impulsantwort" := Ausgangsfolge als Reaktion auf die Impulsfolge



=> liefert vollständige Beschreibung von LTI-Systemen

"linear, time-invariant"

z.B. Verstärker, Filter, usw.

=> auch als lineare Approximation nichtlinearer Systeme

Differenzgleichungen

Berechnung des Ausgangswerts $Y(n)$

- aus gegenwärtigem Eingangswert $X(n)$
- aus früheren Eingangswerten $X(n-k)$
- aus früheren Ausgangswerten $Y(n-k)$ (Rekursion)
- besonders wichtig: linear, konstante Koeffizienten, 2. Ordnung

$$y(n) = A_0 x(n) + A_1 x(n-1) + A_2 x(n-2) - B_1 y(n-1) - B_2 y(n-2)$$

Ordnung N: $B_1 \dots B_N$ kommen vor, $A_1 \dots A_M$, $M < N$

- Übertragungsfunktion $H(z)$ durch Z-Transformation:

$$Y(z) = A_0 X(z) + A_1 X(z)/z + A_2 X(z)/z^2 - B_1 Y(z)/z + B_2 Y(z) / z^2$$

Fouriertransformation

Fouriertransformation:

für periodische, zeit- und wertkontinuierliche Funktionen

auch für periodische, zeit/wert-diskrete Funktionen

Fensterung erzwingt periodische Fortsetzbarkeit

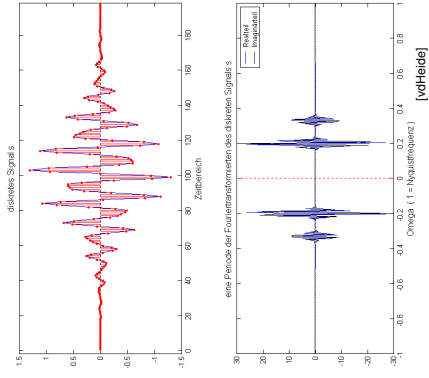
Zahlenfolgen $X(n)$ und Spektrum $X^*(f)$ sind äquivalente Beschreibung von diskreten Signalen

- siehe Matlab-Skript Signalverarbeitung

Fouriertransformation, diskret

- siehe Matlab-Skript Signalverarbeitung [vdHeide]

Zeit- und Frequenzbereich



- Fouriertransformation
- FFT
- Fensterfunktionen
- siehe Matlab-Skript Signalverarbeitung [vdHeide]

Abtasttheorem

nur im Bereich

$$-f_{gs} < f < +f_{gs} \quad (7.8)$$

von null verschieden ist und innerhalb dieses Bereichs mit $U_{\text{per}}(f)$ übereinstimmt, kann für die Zeitfunktion auch

$$u(t) = \int_{-f_{gs}}^{f_{gs}} \left\{ \sum_{n=-\infty}^{\infty} c_n \exp(-j2\pi n \frac{t}{2f_{gs}}) \right\} \exp(j2\pi ft) dt \quad (7.9)$$

geschrieben werden. Man fällt die e-Funktionen zusammen und bildet die Summe der Integrale. So erhält man:

$$u(t) = \sum_{n=-\infty}^{\infty} c_n \int_{-f_{gs}}^{f_{gs}} \exp(-j2\pi n \frac{t}{2f_{gs}}) dt \quad (7.10)$$

Die Lösung des Integrals in dieser Summe ergibt:

$$\int_{-f_{gs}}^{f_{gs}} \exp(-j2\pi n \frac{t}{2f_{gs}}) dt = 2 f_{gs} \frac{\sin(2\pi f_{gs} t \frac{n}{2f_{gs}})}{2\pi f_{gs} t \frac{n}{2f_{gs}}} \quad (7.11)$$

Zur Bestimmung der Fourier-Koeffizienten c_n wird

$$t = k \frac{1}{2 f_{gs}} \quad (7.12)$$

gesetzt. Dann sind in der Summe (7.10) alle Terme mit $n \neq k$ null. Damit gilt für

Abtasttheorem

die Koeffizienten:

$$c_n = \frac{1}{f_{gs}} u\left(n \frac{1}{2 f_{gs}}\right) \quad (7.13)$$

So entsteht aus (7.10) die Interpolationsformel

$$u(t) = \sum_{n=-\infty}^{\infty} u\left(n \frac{1}{2 f_{gs}}\right) \frac{\sin(2\pi f_{gs} t \frac{n}{2f_{gs}})}{2\pi f_{gs} t \frac{n}{2f_{gs}}} \quad (7.14)$$

die aussagt, daß ein bandbegrenztes Signal $u(t)$ mit der Signalgrenzfrequenz f_{gs} vollständig beschrieben ist durch seine Abtastwerte zu den Zeiten

$$t_k = k \frac{1}{2 f_{gs}} \quad k \text{ ganze Zahl} \quad (7.15)$$

Dieses Abtasttheorem für Zeitfunktionen ist von grundlegender Bedeutung für die digitale Signalverarbeitung.

Abtasttheorem

macht die Verarbeitung von Zahlenfolgen überhaupt Sinn ?

Abtasttheorem:

ein auf die halbe Abtastfrequenz (=Nyquistfrequenz) bandbegrenztes Signal kann nach der Abtastung perfekt rekonstruiert werden

- Grundlage aller digitalen Signalverarbeitung
- Beweis mit Fouriertransformation
- siehe [Shannon 48]
- siehe Matlab-Skript Signalverarbeitung

Abtasttheorem

7.4 Abtasttheorem für Zeitfunktionen

Ein reelles bandbegrenztes Signal $u(t)$ hat eine Signalgrenzfrequenz f_{gs} . Für die Fourier-Transformierte $U(f)$ dieses Signals gilt:

$$U(f) = 0 \text{ für } |f| > f_{gs} \quad (7.5)$$

Durch periodische Wiederholung von $U(f)$ mit der Periode $2 f_{gs}$ entsteht eine periodische Frequenzfunktion, die in eine Fourier-Reihe

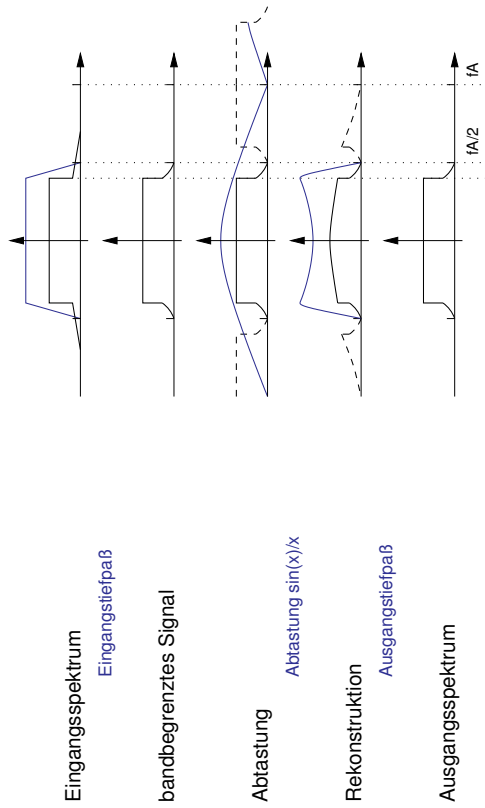
$$U_{\text{per}}(f) = \sum_{n=-\infty}^{\infty} c_n \exp(-j2\pi n \frac{f}{2 f_{gs}}) \quad (7.6)$$

entwickelt werden kann. Die Zeitfunktion

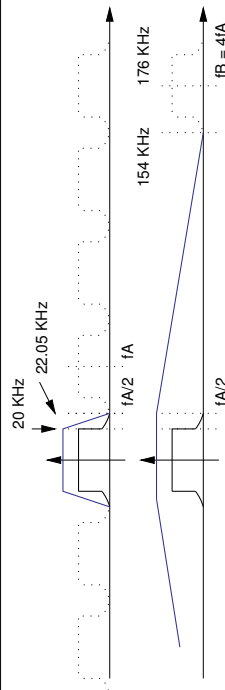
$$u(t) = \int_{-\infty}^{\infty} U(f) \exp(j2\pi ft) df \quad (7.7)$$

ergibt sich als inverse Fourier-Transformierte der Frequenzfunktion $U(f)$. Da diese

Abtasttheorem: Schema



Abtasttheorem: Überabtastung



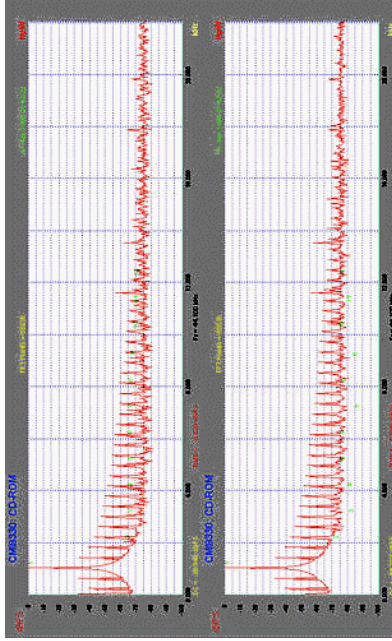
Abtasttheorem:

- vollständige Rekonstruktion des Signals, aber:
- erfordert analogen Tiefpaß sehr hoher Qualität
- Beispiel CD: Dämpfung 96 dB von 20 KHz bis 22.05 KHz
- typische analoge Filter: 6 .. 24 dB / Oktave

Oversampling:

- digitale Neu-Abtastung mit höherer Rate (z.B. 4x, 8x)
- digitale Tiefpaßfilterung beseitigt die Spiegelfrequenzen
- einfacher analoger Tiefpaß genügt

Abtasttheorem: Aliasing-Fehler ...



- Meßwerte der CMI 8330 Soundkarte [et 10/98]
- Sinussignal und massive Anti-Aliasing Fehler (Spiegelfrequenzen)
- massiver Qualitätsverlust durch Stümperei

Quantisierungsfehler

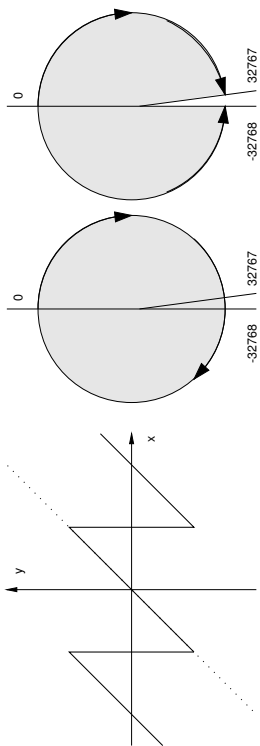
wertdiskret, Zahlendarstellung mit endlicher Wortbreite:

- Quantisierungsfehler von Koeffizienten
- Überlauf bei Additionen
- Auslöschung bei Subtraktionen
- Genauigkeitsverlust bei Multiplikationen
- z.B. andere Lage von Filter-Polen
- dadurch evtl. instabil (!)
- Mathematik: siehe Zölzer, Kap. 2

zeitdiskret, Nyquist-Bedingung

- Vorsicht mit Spiegelfrequenzen

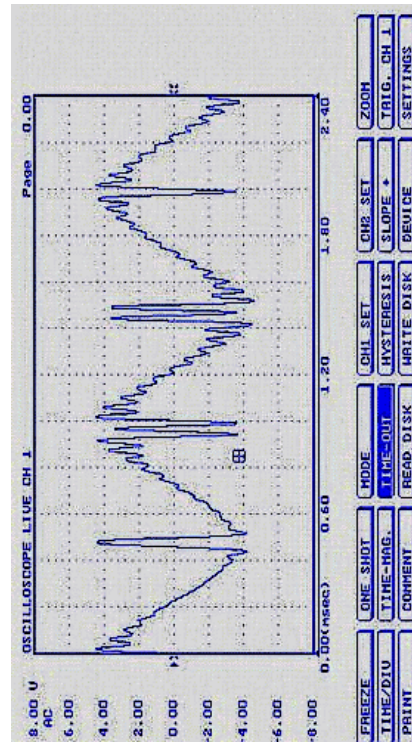
Übersteuerung



was passiert bei Überlauf / Übersteuerung?

- "wraparound" / "saturation" / "exception"
- arithmetische Ausnahmen unerwünscht (z.B. Echtzeitanforderung)
- normale Binärarithmetik liefert wraparound (0, 1, 2, 3, -4, -3, ...)
- => maximaler Fehler (Verzerrungen)
- deshalb "saturation" oft besser für DSP geeignet

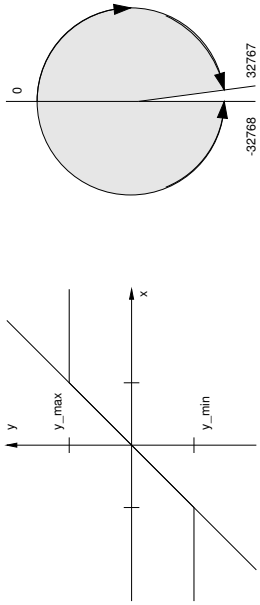
Überlauf: wraparound...



- Beispiel für katastrophalen Überlauf mit wraparound: A/D-Eingang der 24-bit Audiokarte "Event GINA"

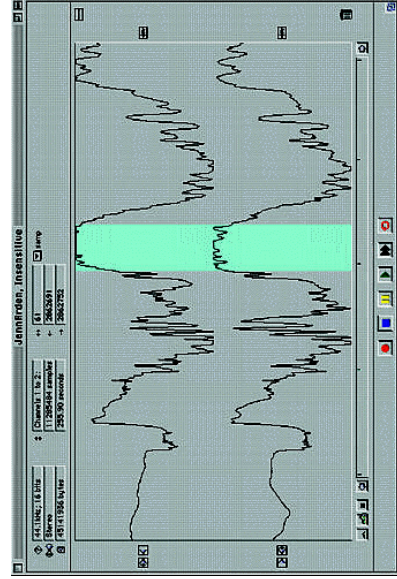
[ct xx/98]

Saturation-Arithmetik



- saturation := "Sättigung"
- Resultat wird auf darstellbaren Zahlenbereich begrenzt
- kein "wraparound", sondern Weiterrechnen mit max/min. Wert
- in vielen Signalprozessoren verfügbar
- auch in MMX

Überlauf: übersteuerte CD



- Beispiel für "saturation" auf übersteuerten CD
- weniger störend als wraparound

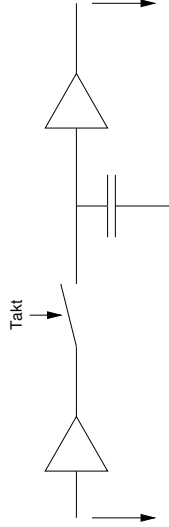
[ct 21/98 120]

A/D-Wandlung

Digitalisierung eines Eingangssignals:

- Zeitquantisierung (Abtaster, Sample&Hold)
- Amplitudenquantisierung (A/D-Konverter)
- Codierung (meistens 2-Komplement)
- nur für bandbegrenzte Eingangssignale
- analoger Tiefpaß vor dem Eingang erforderlich
- diverse Verfahren
- Trade-off zwischen Performance / Genauigkeit / Kosten
- für Audio: 24-bit Wandler bereits erhältlich

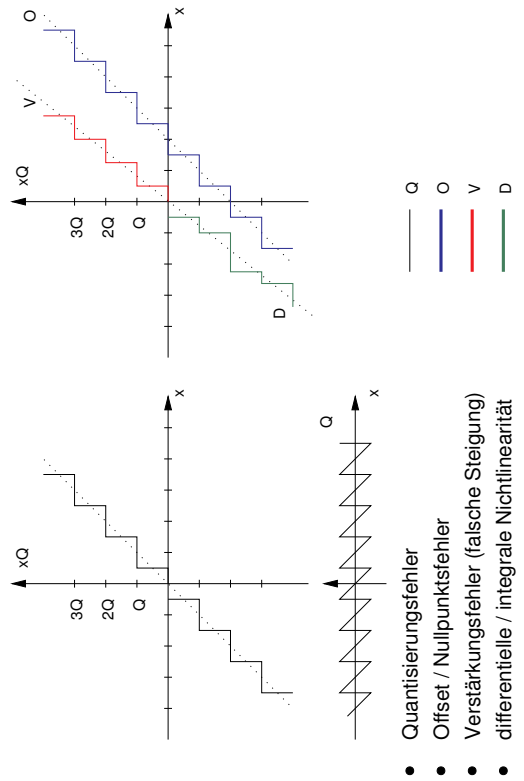
A/D: Abtaster (sample and hold)



Sample: Schalter periodisch schliessen
 Hold: Ladung auf Kondensator gespeichert

- zwei Verstärker zur Entkopplung

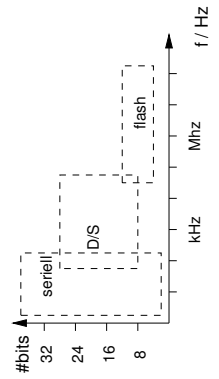
A/D: Parameter



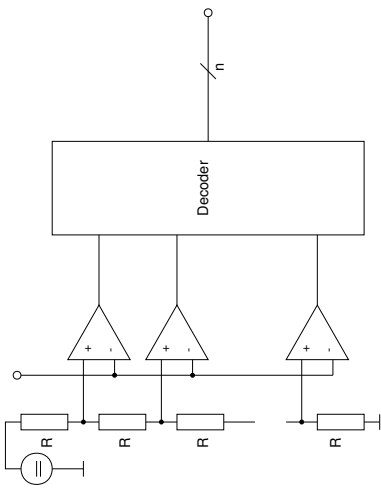
A/D: Verfahren

Vielzahl möglicher Verfahren:

- Parallel-Umsetzung ("Flash") schnell, teuer, < 10 Bit
- serielle Umsetzung langsam, billig, sehr genau
- Sigma-Delta-Umsetzer komplex, genau
- Kombinationen mehrerer Verfahren
- siehe Tietze/Schenk, Zölzer, Datenblätter

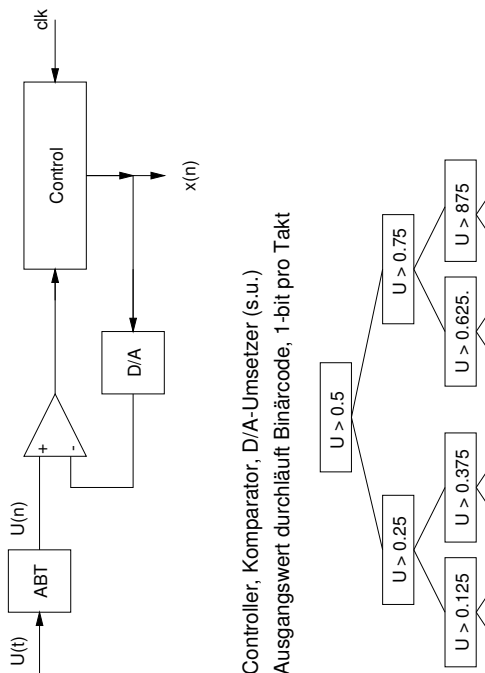


A/D: Parallel-Umsetzer



- erfordert $2^{**}n$ Komparatoren
- sehr schnell
- aber nur für geringe Auflösungen (z.B. 8-bit)

A/D: Wägewerfahren



- Controller, Komparator, D/A-Umsetzer (s.u.)
- Ausgangswert durchläuft Binärcode, 1-bit pro Takt

A/D: Analog Devices AD 1878

PRODUCT OVERVIEW

The AD1878 is a two-channel, 18-bit oversampling ADC based on 2A technology and intended primarily for digital audio applications. The AD1878 is fabricated in a 1.0 μm CMOS process. The AD1878 is fabricated in a 1.0 μm CMOS process. The AD1878 should be read as applying to both parts unless otherwise noted.

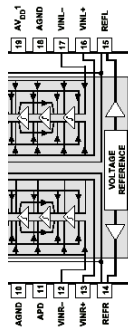
Each input channel of these ADCs is fully differential. Each data converter channel consists of a fifth-order one-bit noise shaping modulator and a digital decimation filter. An on-chip resistor network provides the feedback for the modulator. The AD1878/AD1879 accepts a 256 × F_s input master clock.

Input signals are sampled at 64 × F_s on switched-capacitors. The remaining external sampling amplifiers and modulators are implemented in a 1.0 μm CMOS process. With an input filter, the AD1878/AD1879's proprietary fifth-order differential switched-capacitor modulator architecture shapes the one-bit comparator's quantization noise out of the audio passband. The high order of the modulator randomizes the modulator output, reducing idle tones in the AD1878/AD1879 to very low levels.

The AD1878/AD1879's differential architecture provides increased dynamic range and excellent common-mode rejection characteristics. Because its modulator is single-bit, AD1878/AD1879 is inherently monotonic and has no mechanism for producing differential linearity errors.

The digital decimation filters are single-stage, 4095-tap finite impulse response filters for filtering the modulator's high frequency quantization noise and reducing the 64 × F_s single-bit output data rate to a F_s word rate. They provide linear

*Protected by U.S. Patent Numbers 5655463, 5128653, and others pending.



phase and a narrow transition band that permits the digitization of 20 kHz signals while preventing aliasing into the passband even when using a 44.1 kHz sampling frequency. Passband ripple is less than 0.001 dB, and stopband attenuation exceeds 115 dB.

The flexible serial output port produces data in two-complement, MSB-first format. Input and output signals are in TTL- and CMOS-compatible logic levels. The port is configured by pin selections. The AD1878/AD1879 can operate in either master or slave mode. Each 16-/18-bit output word of a stereo pair can be formatted within a 32-bit field as either right-justified, I²S-compatible, or at user-selected positions. The output can also be truncated to 16-bits by formatting into a 16-bit field.

The AD1878/AD1879 consists of two integrated circuits in a single ceramic 28-pin DIP package. The modulators and reference are fabricated in a BiCMOS process; the decimator and output port, in a 1.0 μm CMOS process. Separating these functions reduces digital crosstalk to the analog circuitry. Analog and digital supply connections are separated to further isolate the analog circuitry from the digital supplies.

The AD1878/AD1879 operates from ±5 V power supplies over the temperature range of -25°C to +70°C.

A/D: Analog Devices AD 1878

AD1878/AD1879*

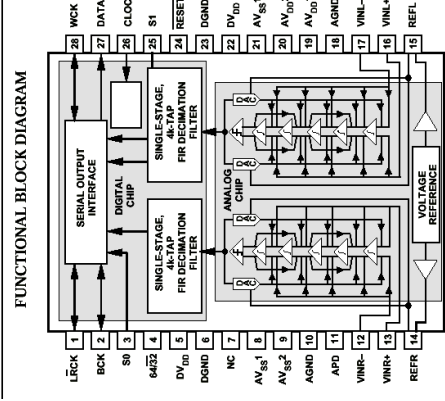


Figure 20. AD1878 S/(THD+N)—1 kHz Tone at -0.5 dBFS (4k-Point FFT)

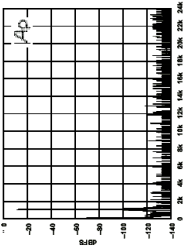


Figure 21. AD1879 S/(THD+N)—1 kHz Tone at -10 dBFS (4k-Point FFT)

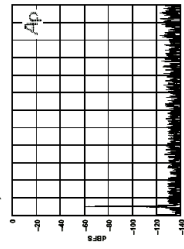
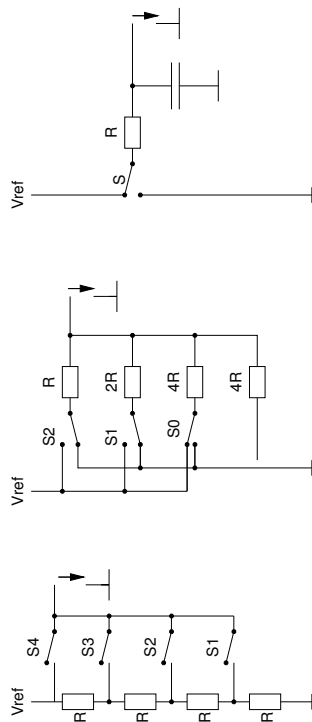


Figure 22. AD1879 S/(THD+N)—1 kHz Tone at -50 dBFS (4k-Point FFT)

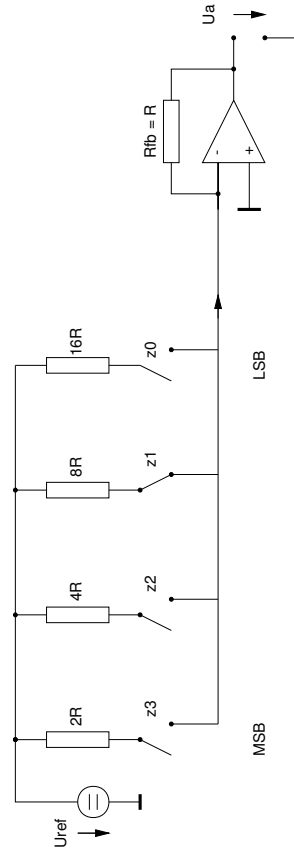
D/A-Wandlung

D/A: Umwandlung numerischer Werte in analoge Werte (Spannungen)

- Parallelverfahren
- Wägetverfahren
- Zählverfahren

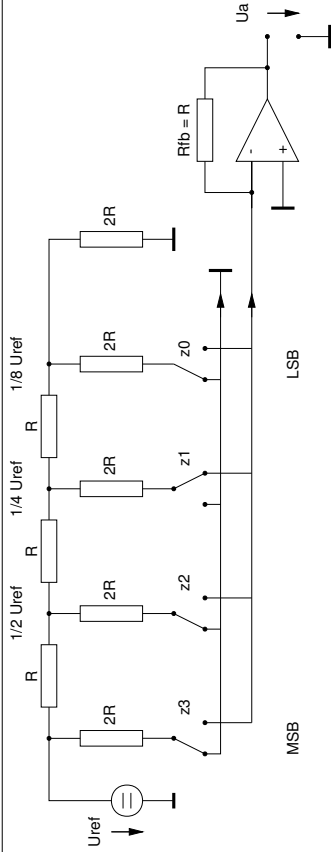


D/A: Widerstandsnetzwerk



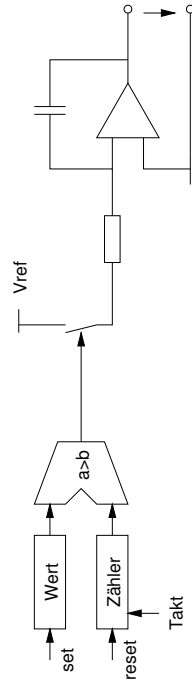
- Summation gewichteter Ströme (Kirchhoff)
- $$U_a = -(U_{ref}/16R) * (8z_3 + 4z_2 + 2z_1 + z_0)$$
- erfordert Widerstände mit hoher Genauigkeit
 - datenabhängige Belastung von Uref
- => kaum realisierbar

D/A: R-2R Netzwerk



- VLSI: Herstellung unterschiedlicher Widerstände schwierig
 - aber gleiche Widerstände mit hoher Genauigkeit
- => fortgesetzte Spannungsteilung mit R-2R Netzwerk
- $$U_a = -U_{ref} * (R_{fb}/16R) * (8z_3 + 4z_2 + 2z_1 + z_0)$$
- in CMOS-Technologie oft verwendet

D/A: serieller Umsetzer



- "Pulsweitenmodulation"
- Kondensator wird geladen, solange Wert > Zählerstand
- Taktrate des Zählers: 2**n fache der Samplefrequenz
- langsam
- geringer (analoger) Hardwareaufwand

Quantisierungsprobleme

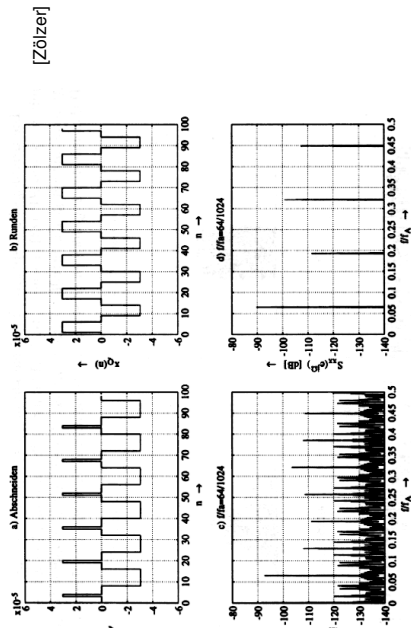
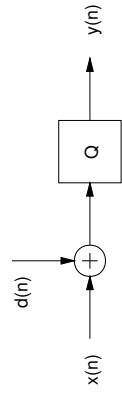


Bild 2.19: 1 Bit Aussteuerung des Quantisierers bei Abschneiden a/c und Runden b/d

- Quantisierung erzeugt störende "Nebensignale"
- Verbesserung durch Hinzufügen von Rauschen: "Dithering"

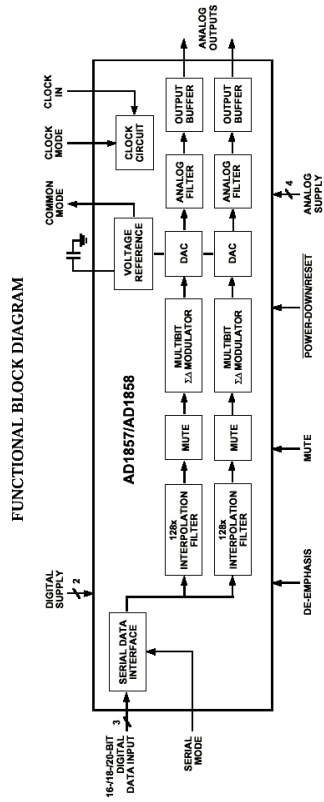
Dither: Prinzip



bits
 $w+r$
 $s < r$
 w
 x(n) Eingangssignal
 d(n) Zufallsignal (dither)
 y(n) Ausgangssignal

- Q: Re-Quantisierung des Signals, oder D/A-Wandler
- Amplitude des Dithers kleiner als LSB des Eingangssignals
- Verteilung (Statistik) des Dithers frei wählbar: noise shaping
- Mathematik siehe Zöizer S.38 ff

D/A: Beispiel AD 1858



- Sigma-Delta Verfahren

D/A: Beispiel AD 1858

ANALOG PERFORMANCE		Min	Typ	Max	Units
AD1857 Resolution			18		Bits
AD1858 Resolution			16		Bits
Dynamic Range (20 Hz to 20 KHz, -60 dB Input)					dB
No A-Weight Filter			91		dB
With A-Weight Filter			-90	-85	dB
Total Harmonic Distortion + Noise			0.003	0.006	%
Analog Outputs					V p-p
Single-Ended Output Range (± Full Scale)		2.8	3.0	3.2	V
Output Impedance at Each Output Pin			<200		Ω
Output Capacitance at Each Output Pin			20		pF
Out-of-Band Energy (0.3 × F _s to 100 KHz)			-72.5		dB
CMOUT		2.1	2.25	2.4	V
DC Accuracy			±3.0		%
Gain Error			±0.1	±7.5	dB
Interchannel Gain Mismatch			±0.2		ppm/°C
Gain Drift			130	300	dB
Interchannel Crosstalk (EIA method)			-120	-100	dB
Interchannel Phase Deviation			±0.1		Degrees
Mute Attenuation			-100	-90	dB
De-emphasis Gain Error				±0.1	dB

- derzeit bis 24-bit Wandler erhältlich

Dithering: Varianten, Realisierung

- gleichverteilte Zufallszahlen $d(n)$

$$\begin{aligned} \text{RECT} &= d_1(n) \\ \text{TRI} &= d_1(n) + d_2(n) \\ \text{Hochpaß} &= d_1(n) - d_1(n-1) \end{aligned}$$
- Wertebereich $|d_{\text{RECT}}| = Q/2$
Wertebereich $|d_{\text{TRI}}| = Q$
- höhere Rauschleistung des TRI/HP-Dithers
- aber besserer Klang
- Details: Zölzer S. 38 ff
- "noise shaping" für psychoakustisch optimales Dithering

Dither: bei 1 bit Aussteuerung

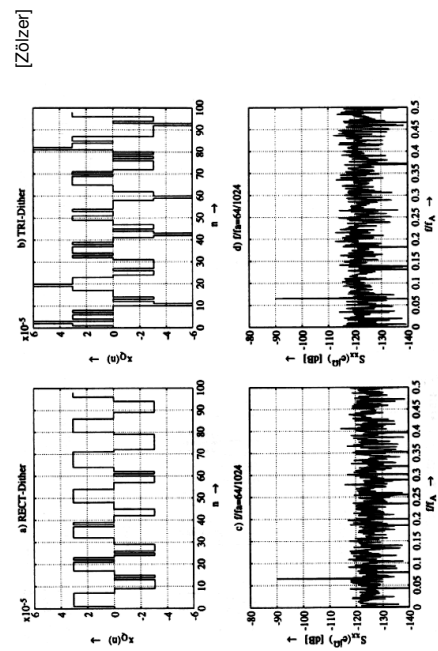


Bild 2.20: 1 Bit Aussteuerung - Rundungskennlinie mit RECT-Dither a/c und TRI-Dither b/d

Dither: bei 0.25 bit Aussteuerung

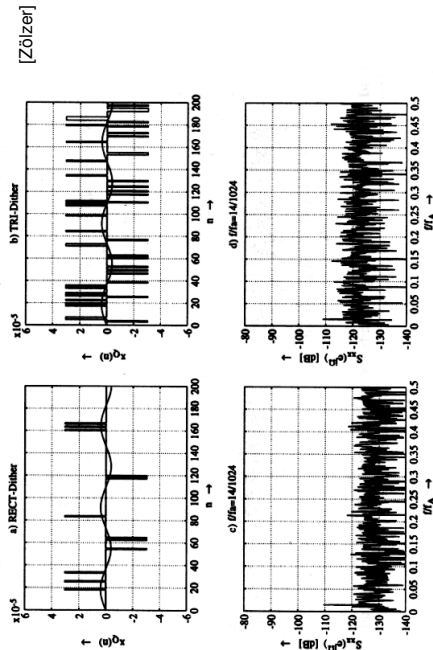


Bild 2.21: 0.25 Bit Aussteuerung - Rundungskennlinie mit RECT-Dither a/c und TRI-Dither b/d

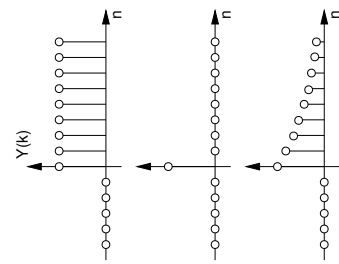
Dither: Noise-Shaping

- ungleichmäßiger Verlauf der Hörschwelle
- => Idee: Verteilung des Rauschens an Hörschwelle anpassen
- möglichst wenig Rauschen bei ca. 4 KHz
- dafür höheres Rauschen bei tiefen/hohen Frequenzen
- höherer Rechenaufwand für das Dithering
- aber Gewinn von ca. 2.5 bis 6 Bit an Auflösung (!)
- z.B. effektiv 20 bit mit 18 bit D/A Wandlern
- auch beim CD-Mastering: z.B. Sony Super Bit Mapping
- Mathematik: siehe Zölzer, Kapitel 2.3

Digitale Signalverarbeitung (2)

- Motivation
- analoge vs. digitale Signalverarbeitung
- Literatur
- Zahlenfolgen, LTI-Systeme
- Abtasttheorem
- Spektrum, FFT, Übertragungsfunktion
- Quantisierung, AD/DA Konverter, Dither
- Z-Transformation
- digitale Filter
- Dynamikbeeinflussung
- Raumsimulation, Hall

Zahlenfolgen, Elementarfolgen



Wertemenge $\{Y(k)\}$
Zuordnung: Index $k \rightarrow Y(k)$

wichtige "Elementarfolgen":

- Impulsfolge
- Sprungfolge
- Exponentialfolge
- Sinusfolge

lineare Systeme:

- Signale aus Elementarfolgen zusammensetzen
- Systemverhalten entsprechend berechenbar

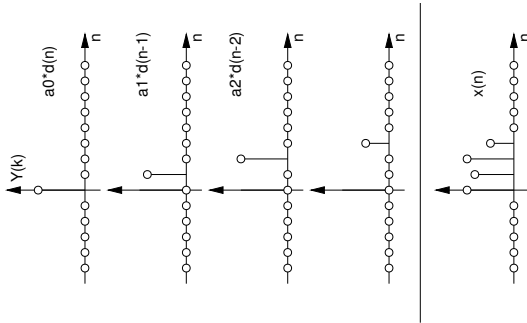
Impulsfolge:

Konstruktion beliebiger Zahlenfolgen:

- Summe
- skaliertes
- zeitverschobener Impulsfolgen

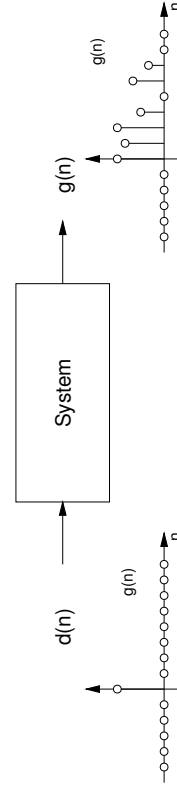
$$x(n) = a_0 \cdot d(n) + a_1 \cdot d(n-1) + \dots$$

- gleichwertig: Konstruktion mit
- skalierten, zeitverschobenen
 - Sprungfolgen



Impulsantwort

"Impulsantwort" := Ausgangsfolge als Reaktion auf die Impulsfolge



=> liefert vollständige Beschreibung von LTI-Systemen

"linear, time-invariant"

z.B. Verstärker, Filter, usw.

=> auch als lineare Approximation nichtlinearer Systeme

Differenzgleichungen

Berechnung des Ausgangswerts $Y(n)$

- aus gegenwärtigem Eingangswert $X(n)$
- aus früheren Eingangswerten $X(n-k)$
- aus früheren Ausgangswerten $Y(n-k)$ (Rekursion)
- besonders wichtig: linear, konstante Koeffizienten, 2. Ordnung

$$y(n) = A_0 x(n) + A_1 x(n-1) + A_2 x(n-2) - B_1 y(n-1) - B_2 y(n-2)$$

Ordnung N : $B_1 \dots B_N$ kommen vor, $A_1 \dots A_M, M < N$

- Übertragungsfunktion $H(z)$ durch Z-Transformation:

$$Y(z) = A_0 X(z) + A_1 X(z)/z + A_2 X(z)/z^2 - B_1 Y(z)/z + B_2 Y(z)/z^2$$

$$H(z) = Y(z) / X(z)$$

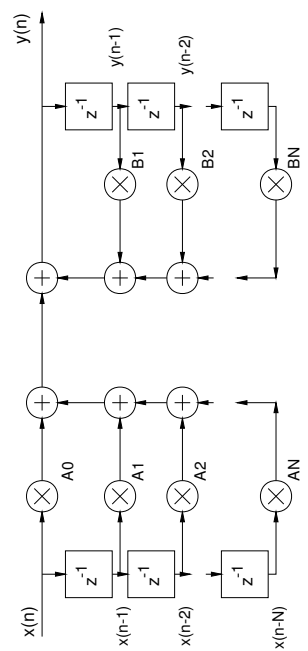
DGL: Direktform

Differenzgleichungen, linear, konstante Koeffizienten

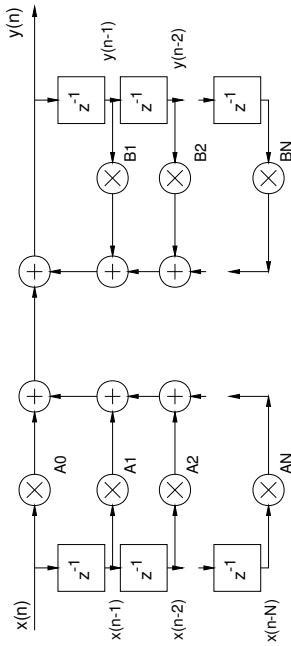
$$y(n) = A_0 x(n) + A_1 x(n-1) + A_2 x(n-2) - B_1 y(n-1) - B_2 y(n-2)$$

Umsetzung in "Direktform":

- Addition, Multiplikation, Verzögerung (=Register)

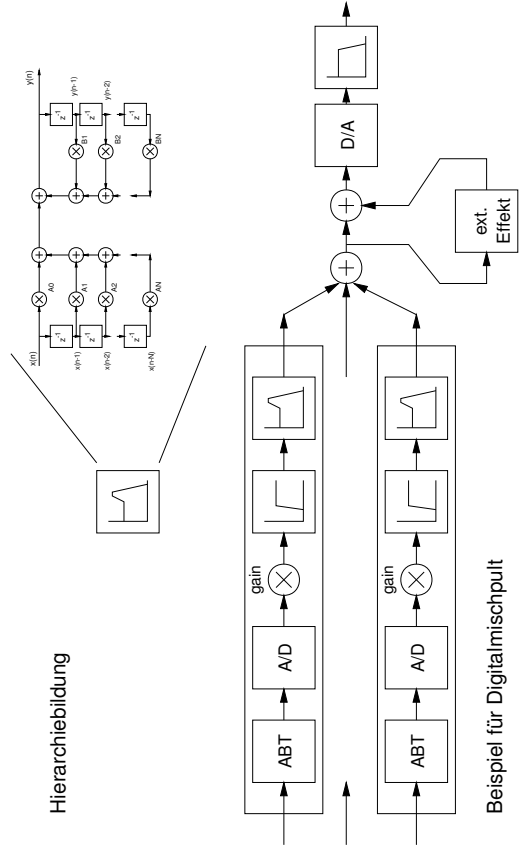


DGLs: Blockschaltbilder



- Direktform erlaubt sofortige Realisierung von DSP-Algorithmen
 - Varianten möglich (Vertauschen von Registern/Operatoren)
 - bei Bedarf zusätzlich weitere (z.B. nicht-lineare) Operatoren
- => Darstellung mit Blockschaltbildern

Blockschaltbilder:



Beispiel für Digitalmischpult

Z-Transformation

Einer Zahlenfolge $x(n)$ wird durch die unendliche Summe

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n) z^{-n} \tag{7.3}$$

eine gebrochene rationale Funktion $X(z)$ zugeordnet. Beim praktischen Gebrauch dieser Transformation spielen Korrespondenztabelle eine wichtige Rolle. Man strebt an, die Transformation vom Folgebereich in den z -Bereich sowie auch in umgekehrter Richtung mit Hilfe der Tabellen durchzuführen. Dabei soll wieder zwischen Korrespondenzen für die Zahlenfolgen und Korrespondenzen für die Rechenoperationen unterschieden werden.

In der Regel wird die Z-Transformation als einseitige Transformation durchgeführt. Dabei werden kausale Zahlenfolgen transformiert, d.h. Folgen, deren Elemente für $n < 0$ verschwinden.

- Umwandlung Differenzengleichung \rightarrow algebra. Gleichung

Z-Transformation: Tabelle

- Berechnung via Summation der Reihe oder aus Tabellen
- Rücktransformation via Partialbruchzerlegung und Tabellen

In den periodischen Funktionen der nachfolgenden Tabelle ist t_0 eine bezogene, dimensionslose Frequenz, die angibt, wieviele Folgeelemente auf eine Periode der Funktion entfallen.

Folgebereich	z-Bereich
$f(n)$	$F(z)$
$\delta(n)$	1
$\varepsilon(n)$	$\frac{z}{z-1}$
$\varepsilon(n) e^{\alpha n}$	$\frac{z}{z-e^\alpha}$

Z-Transformation: Tabelle

Folgebereich	z-Bereich
$\varepsilon(n) z^{2\pi t_0 n}$	$\frac{z}{z - e^{j2\pi t_0}}$
$\varepsilon(n) \sin(2\pi t_0 n)$	$\frac{z \sin(2\pi t_0)}{z^2 - 2z \cos(2\pi t_0) + 1}$
$\varepsilon(n) \sin(2\pi t_0 n + \varphi)$	$\frac{z^2 \sin \varphi + z \sin(2\pi t_0 - \varphi)}{z^2 - 2z \cos(2\pi t_0) + 1}$
$\varepsilon(n) \cos(2\pi t_0 n)$	$\frac{z [z - \cos(2\pi t_0)]}{z^2 - 2z \cos(2\pi t_0) + 1}$
$\varepsilon(n) e^{\alpha n} \sin(2\pi t_0 n + \varphi)$	$\frac{z \sin \varphi + e^{\alpha} \sin(2\pi t_0 - \varphi)}{z^2 - 2z e^{\alpha} \cos(2\pi t_0) + e^{2\alpha}}$
$\varepsilon(n) e^{\alpha n} \cos(2\pi t_0 n)$	$\frac{z [e^{-\alpha} \cos(2\pi t_0)]}{z^2 - 2z e^{\alpha} \cos(2\pi t_0) + e^{2\alpha}}$
$\varepsilon(n) n$	$\frac{z}{(z-1)^2}$
$\varepsilon(n) \frac{1}{2} n^2$	$\frac{z(z+1)}{2(z-1)^3}$
$\varepsilon(n) n e^{\alpha n}$	$\frac{z e^{\alpha}}{(z - e^{\alpha})^2}$

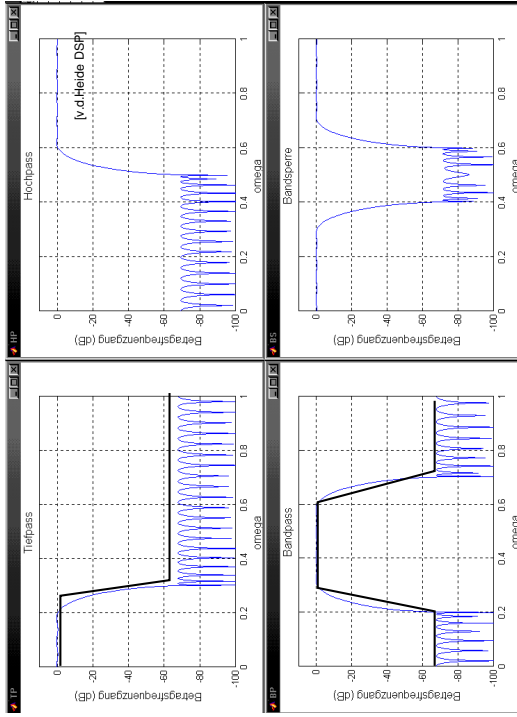
Z-Transformation: Eigenschaften

7.2.3 Korrespondenzen der Rechenoperationen

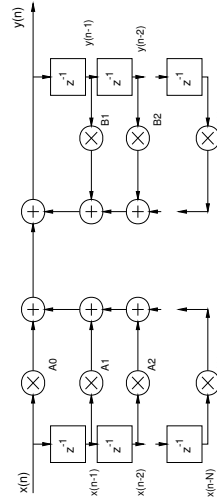
$f(n) = 0$ für $n < 0$

Folgebereich	z-Bereich	Namen
$f_1(n) + c_2 f_2(n)$	$c_1 F_1(z) + c_2 F_2(z)$	Linearität
$f(n - N)$	$z^{-N} F(z)$	Zeitverschiebung
$f_1(n) * f_2(n)$	$F_1(z) F_2(z)$	Faltung
$n f(n)$	$-z \frac{dF(z)}{dz}$	
$e^{\alpha n} f(n)$	$F\left(\frac{z}{e^\alpha}\right)$	Skalierung
$f(n) - f(n-1)$	$\frac{z-1}{z} F(z)$	Zeitdifferenz
$\sum_{k=0}^n f(k)$	$\frac{z}{z-1} F(z)$	Zeitmittelwert

Filter: Tiefpaß, Hochpaß, etc.



Filter: Frequenzgang



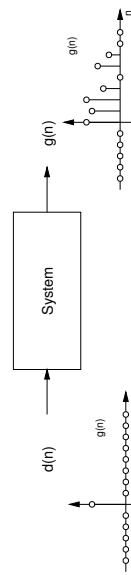
- normierte Frequenz $F = f/FA$
- direkte Berechnung des Frequenzgangs via $H(z)$:

$$|A(j\omega)| = \sqrt{\left[\sum_{k=0}^n \alpha_k \cos 2\pi k F \right]^2 + \left[\sum_{k=0}^n \alpha_k \sin 2\pi k F \right]^2} + \sqrt{\left[\sum_{k=0}^n \beta_k \cos 2\pi k F \right]^2 + \left[\sum_{k=0}^n \beta_k \sin 2\pi k F \right]^2}$$

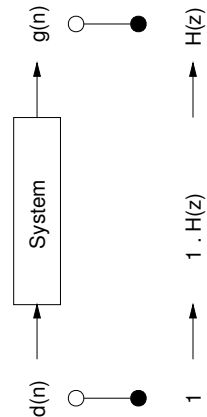
Z-Transformation: Beispiel

- Differenzgleichung zweiter Ordnung:
 $y(n) = A_0 x(n) + A_1 x(n-1) + A_2 x(n-2) - B_1 y(n-1) - B_2 y(n-2)$
- Linearität: gliedweise Transformation in den z-Bereich
- $Y(z) = A_0 X(z) + A_1 X(z)/z + A_2 X(z)/z^2 - B_1 Y(z)/z + B_2 Y(z)/z^2$
- Übertragungsfunktion $H(z) := Y(z) / X(z)$
- $H(z) = \frac{A_0 + A_1/z + A_2/z^2}{1 + B_1/z + B_2/z^2}$
- Realisierung der Zeitverzögerung $1/z$: einfaches Register

Impulsantwort - Übertragungsfunktion

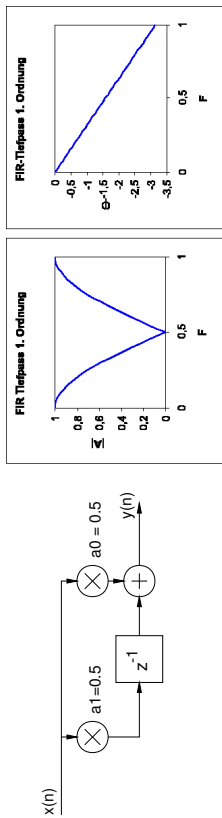


Beziehungen: Zeitbereich / Z-Bereich:



- Ausgangsfolge $y(n)$: als Faltung $x(n) * g(n)$

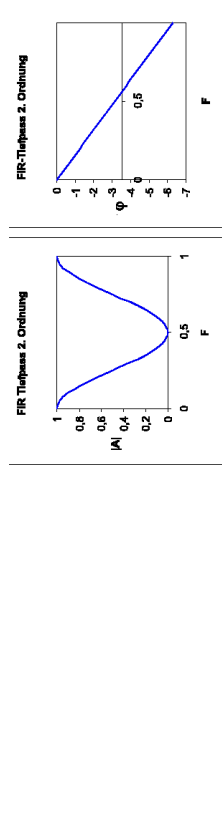
Filter: FIR-Tiefpass



"minimale" Filterstruktur:

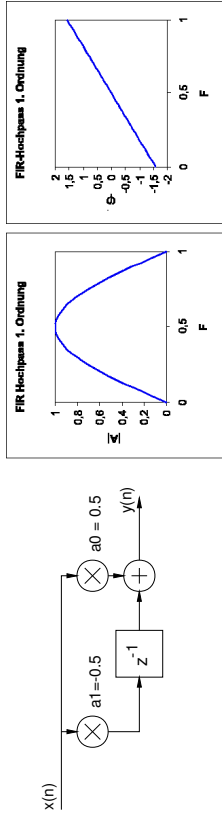
- zwei Koeffizienten, keine Rückkopplung
- Antwort auf $x(n) = 1, 1, 1, 1, 1, \dots$
 $y(n) = 1, 1, 1, 1, 1, \dots$
- Antwort auf $x(n) = 1, -1, 1, -1, 1, \dots$
 $y(n) = 0, 0, 0, 0, 0, \dots$

Filter: FIR-Tiefpass



- Antwort auf $x(n) = 1, 1, 1, 1, 1, \dots$
 $y(n) = 1, 1, 1, 1, 1, \dots$
- Antwort auf $x(n) = 1, -1, 1, -1, 1, \dots$
 $y(n) = 0, 0, 0, 0, 0, \dots$

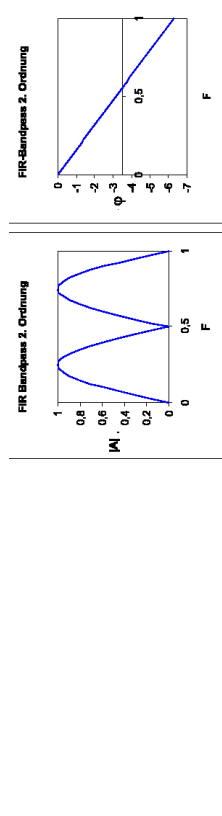
Filter: FIR-Hochpass



"minimale" Filterstruktur:

- zwei Koeffizienten, keine Rückkopplung
- Antwort auf $x(n) = 1, 1, 1, 1, 1, \dots$
 $y(n) = 0, 0, 0, 0, 0, \dots$
- Antwort auf $x(n) = 1, -1, 1, -1, 1, \dots$
 $y(n) = 1, 1, 1, 1, 1, \dots$

Filter: FIR-Bandpass



- Antwort auf $x(n) = 1, 1, 1, 1, 1, \dots$
 $y(n) = 0, 0, 0, 0, 0, \dots$
- Antwort auf $x(n) = 1, -1, 1, -1, 1, \dots$
 $y(n) = 0, 0, 0, 0, 0, \dots$

Filter: FIR - IIR

zwei fundamentale Filterarchitekturen: (siehe v.d.Heide DSP-VL)

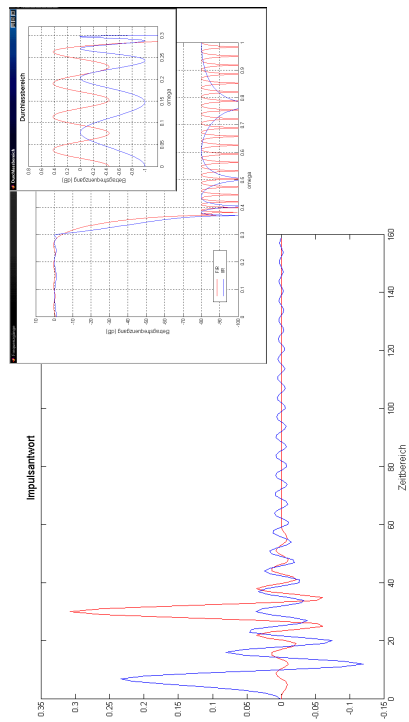
FIR, "finite impulse response"

- nichtrekursive Filter, $y(n) = \text{Summe } A[k] * x(n-k)$
- analog nicht realisierbar
- Ausgangssignal um $N(N/2)$ Perioden verzögert
- Frequenz- und Phasengang separat wählbar

IIR, "infinite impulse response"

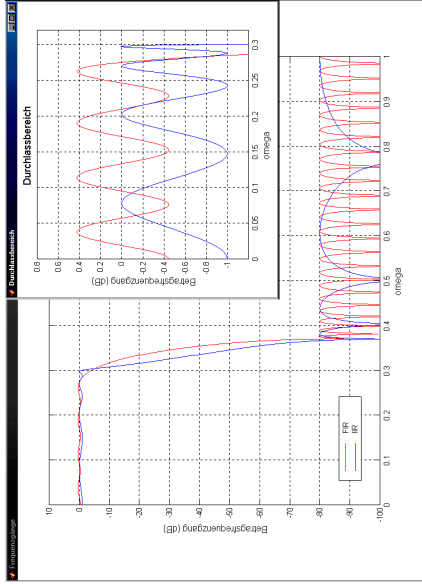
- rekursive Realisierung, $y(n)$ abhängig von $x(n-k)$ und $y(n-m)$
- Impulsantwort (im Prinzip) unendlich lang
- benötigt weniger Koeffizienten als "gleichwertiges" FIR-Filter
- evtl. Stabilitätsproblem (bzw.: resonanzfähig) (z.B. durch Quantisierung der Koeffizienten eines stabilen Filters)

Filter: Impulsantwort bei FIR / IIR



- Maximum der Impulsantwort bei FIR später als bei IIR (mehr Koeff.)
- sehr lange (infinite) Impulsantwort der IIR-Filters

Filter: Frequenzgang bei FIR / IIR



- "Ripple" im Durchlaß- und Sperrbereich
- Steilheit und Sperrdämpfung

Filter: Berechnung

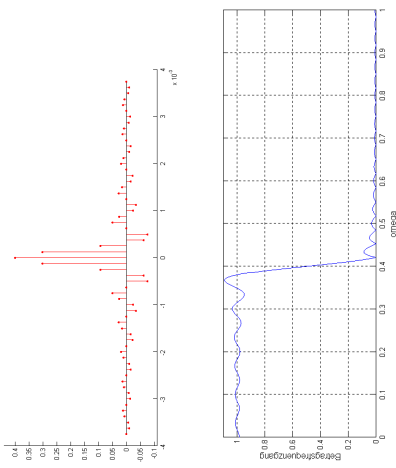
Eigenschaften der Filter voll durch die Koeffizienten A, B bestimmt

- diverse Algorithmen zur Berechnung der A, B
- abhängig von "Nebenbedingungen"

Frequenzgang möglichst

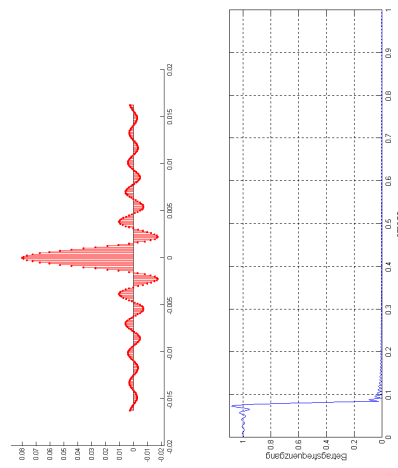
- glatt
- steil
- minimaler "ripple" im Durchlaß- oder Sperrbereich
- etc.
- siehe Matlab-Demo Filterkennlinien

Filter: Anzahl der Koeffizienten (FIR)



- geringe Anzahl der Koeffizienten
- Frequenzgang nicht sehr steil

Filter: Anzahl der Koeffizienten (FIR)



- geringe Anzahl der Koeffizienten
- Frequenzgang wesentlich steiler, immer noch "ripple"

Filter: Klangregelung

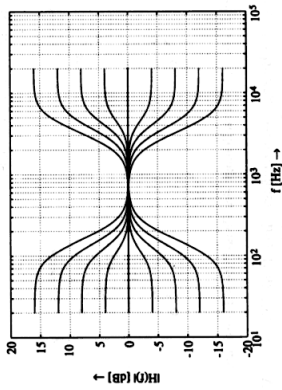


Bild 5.8: Frequenzgänge eines Tiefen/Höhen-Shelving-Filters - Tiefen-Shelving-Filter $f_0 = 100$ Hz (Parameter V_0), Höhen-Shelving-Filter $f_0 = 5000$ Hz (Parameter V_0)

- "Shelving"-Filter zur Klangregelung Höhen/Bässe
- zwei Parameter: Trennfrequenz und Verstärkung/Dämpfung
- Hervorheben einzelner Instrumente / einzelner Nuancen
- Hörbeispiele mit Erläuterungen: "keyboards mix-tricks" Serie

Filter: Koeffizienten

Transformation in den Z-Bereich. Zur Realisierung eines digitalen Filters wird die im S-Bereich entworfene Übertragungsfunktion $H(s)$ mit Hilfe einer geeigneten Transformation in eine Übertragungsfunktion $H(z)$ überführt. Die linksinvariante Transformation ist nicht geeignet, da sie bei einer nicht auf die halbe Abtastfrequenz bandbegrenzten Übertragungsfunktion $H(s)$ auf Überlappungseffekte im Frequenzgang führt. Eine unabhängige Transformation von Pol- und Nullstellen von der S-Ebene in Pol- und Nullstellen in der Z-Ebene ist mit Hilfe der bilinearen Transformation

$$s = \frac{2}{T} \frac{z-1}{z+1} \tag{5.16}$$

möglich. Die folgende Tabelle 5.2 enthält die Koeffizienten der Übertragungsfunktion 2. Ordnung

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{5.17}$$

die mit der bilinearen Transformation und der Hilfsgröße $K = \tan(\omega_0 T/2)$ für die verschiedenen Filtertypen bestimmt sind. Strategien zur zeitvarianten Umschaltung von Audio-Filtern finden sich in [Zol93].

Filter: Koeffiziententabelle

Tabelle 4.2: Audio-Filterkoeffizienten

Tiefpass (2. Ordnung)		Hochpass (2. Ordnung)	
a1	a2	b1	b2
$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$
$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$
Tiefenabstufung (Abstufung $\omega_c = 10^{0.05 \cdot \text{dB}}$)			
a1	a2	b1	b2
$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$
$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$
Höhenabstufung (Abstufung $\omega_c = 10^{0.05 \cdot \text{dB}}$)			
a1	a2	b1	b2
$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$
$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$
Höhenabstufung (Abstufung $\omega_c = 10^{0.05 \cdot \text{dB}}$)			
a1	a2	b1	b2
$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$
$\frac{1 + \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 - \cos(\omega_c)}{2}$	$\frac{1 + \cos(\omega_c)}{2}$

- alle üblichen Audiofilter

Dynamik-Bearbeitung

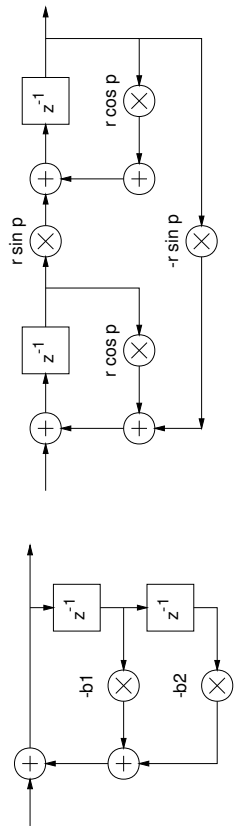
Anpassen der Dynamik (Pegeldifferenzen):

- Schutz vor Übersteuerung, z.B. von A/D-Wandlern "limiter"
- Unterdrückung kleiner Störsignale "noise gate"
- Anpassen großer Pegeldifferenzen an schlechteres System
- Optimierung der Lautheit (insb. Popmusik) "compressor"
- Anpassen an Umgebungseinflüsse
- Lautstärkeregelung im Auto

Filter: "Klang" digitaler Filter ?!

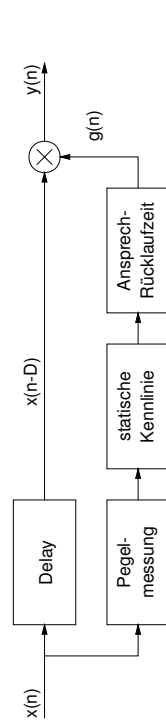
Klangvariationen digitaler Filter? ja, abhängig von

- Algorithmus (FIR, IIR)
- Architekturvarianten
- Wortbreiten, Rundungsfehler



(Zölzer S. 148ff)

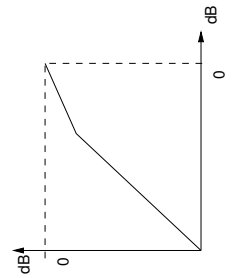
Dynamikbearbeitung: Blockschaltbild



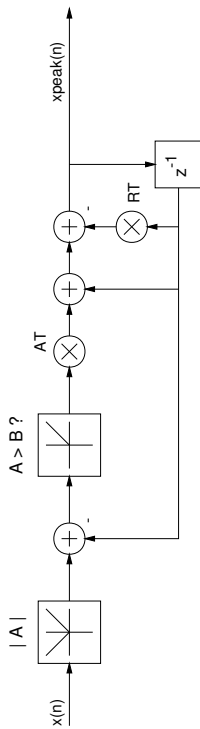
analog: ohne Delay, daher Probleme bei schnellen Änderungen

$$\text{Kompressionsfaktor } R = \frac{dPx}{dPy}$$

- unendlich $R > 1$
- Compressor $0 < R < 1$
- Expander $R = 0$
- Noisegate



Dynamik: Pegelmessung

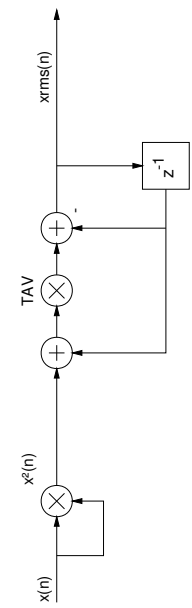


- Spitzenwertmessung:
 AT attack time
 RT release time

$$H(z) = \frac{AT}{1 - (1 - AT - RT)z^{-1}}$$

$$x_{peak}(n) = (1 - AT - RT) * x_{peak}(n-1) + AT * |x(n)|$$

Dynamik: Mittelwert Pegelmessung

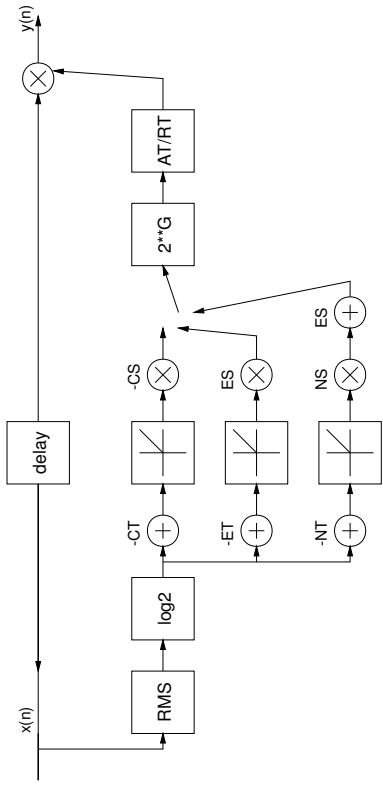


- Mittelwertmessung:

$$H(z) = \frac{TAV}{1 - (1 - TAV)z^{-1}}$$

$$x_{rms}(n) = (1 - TAV) * x_{rms}(n-1) + TAV * x^2(n)$$

Dynamik: Kompressor / Expander / Noisegate



- Schwellwertvergleich, dann Entscheidung K/E/N

Kompression: Beispiel

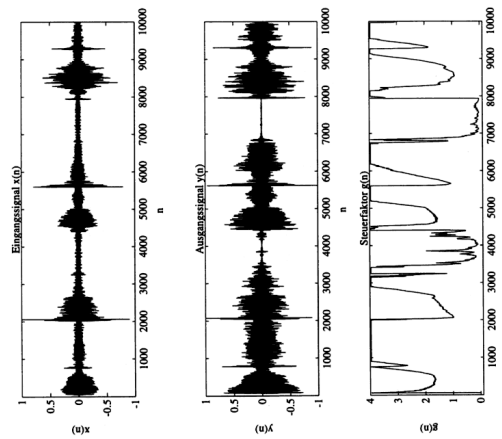
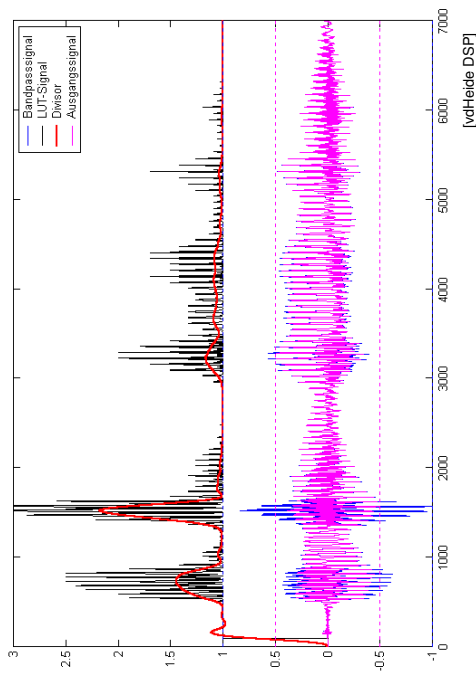


Bild 7.11: Zeigsigale $x(n)$, $y(n)$ und $g(n)$ der Dynamikbeeinflussung [Zöbier]

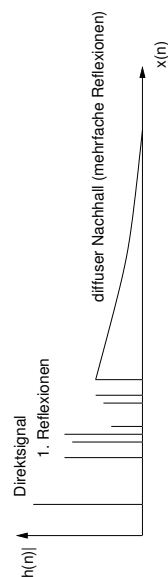
Kompression: Beispiel



Raumsimulation, Hall

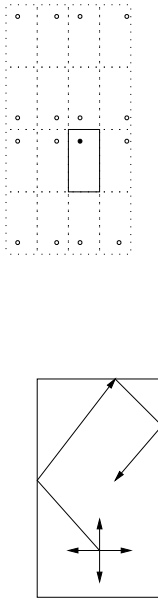
Raumsimulation := künstliche Erzeugung einer Raumakustik

- Originalsignal
- Reflexionen an Wänden / anderen Objekten
- typischer Verlauf:



- Originalsignal, erste und mehrfache Reflexionen
- Klangverfärbung, insb. Höhendämpfung

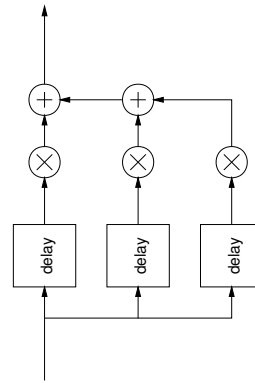
Hall: Box-Modell



"Strahlen-Modell" "Spiegelquellen"

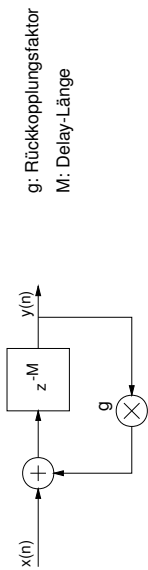
- mathematisches Modell eines Raums (als Übertragungsfunktion)
 - in einfachen Fällen (Box) exakt lösbar
 - ansonsten numerische Näherung
 - aber: "echter" Hall ist extrem aufwendig
- => Suche nach vereinfachten Algorithmen

Hall: erste Reflexionen



- direktes Erzeugen der wichtigen ersten Reflexionen
- mit mehrfacher Verzögerung
- evtl. auch Stereo / räumlich differenziert
- "Räumlichkeit" := psychoakustische Bewertung der Reflexionen
- angenehme Werte abhängig vom Signal (Klassik Pop, Sprache, ...)

Hall: diffuser Nachhall



g: Rückkopplungsfaktor
M: Delay-Länge

Erzeugen des "diffusen" Hallanteils ?!

- zusammengesetzt aus sehr hoher Anzahl von Teilreflexionen
 - direkte Simulation per delay/Faktor zu aufwendig
 - Vereinfachungen notwendig
- => rekursive Filter, z.B. Kammfilter / Allpaßfilter [Schroeder 61]
- mit exponentiellem Abfall der Impulsantwort
 - aber keine "Verdichtung" des Hallsignals

Hall: Kammfilter / Allpaßfilter

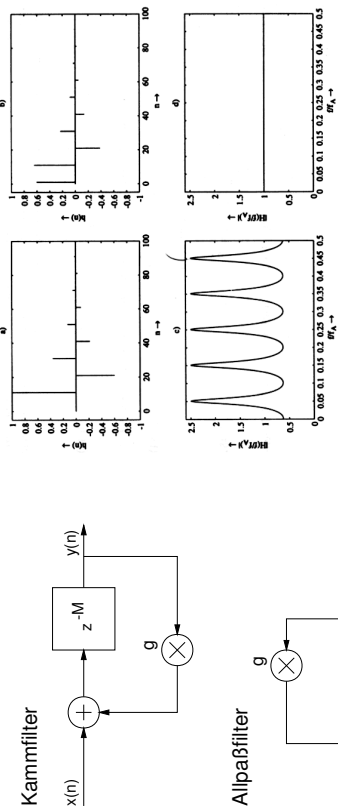


Bild 6.14: a) Impulsantwort Kammfilter (M=10, g=0.6), b) Impulsantwort Allpaßfilter (M=10, g=0.6), c) Betragfrequenzgang Kammfilter, d) Betragfrequenzgang Allpaßfilter

- jeweils Impulsantwort
- und Frequenzgang
- exp. Abklingen der Impulsantwort

Hall: diffuser Nachhall

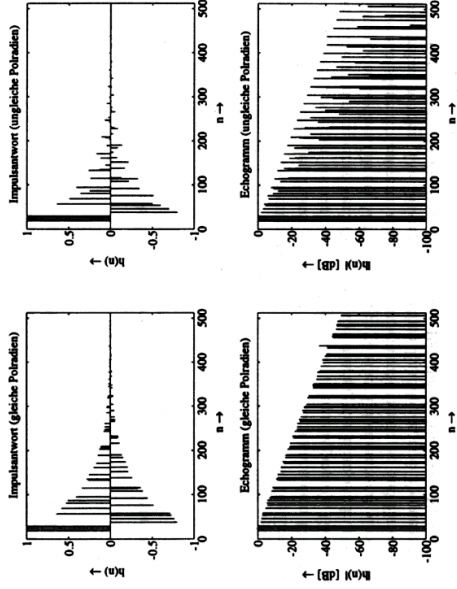
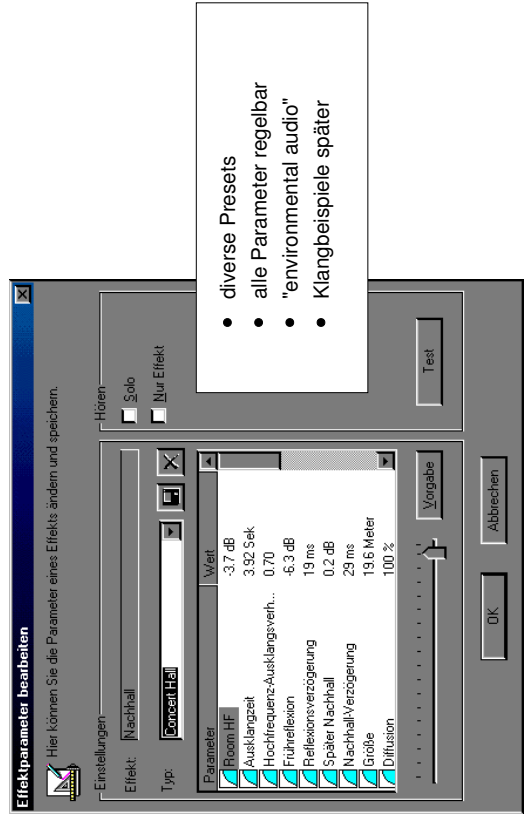


Bild 6.19: Impulsantwort und Echogramm (mehrere Kammfilter parallel)

Hall: Soundblaster Live



- diverse Presets
- alle Parameter regelbar
- "environmental audio"
- Klangbeispiele später

Audioqualität: transparent channel

- analog (LP) besser als digital (CD, DAT) ?!
 - Übertragungsqualität analysieren
- => Anforderungen an einen "transparenten Kanal" ?

J. R. Stuart: "Coding high quality audio"
www.meridian.co.uk/ara/index.html

Thesen:

- CD und DAT knapp unzureichend
- vor allem höhere Abtastrate nötig
- DVD-Audio / SACD ist Verschwendung
- bei Musikproduktion auf Quantisierungsfehler vermeiden
- Dithering/Noiseshaping unbedingt notwendig

bits	f / KHZ
16	44.1 / 48
16	66
24	96

Motivation: "CD-Qualität"

As digital audio has progressed, we have also evolved the capability to record and play back with resolution that exceeds that of Red Book CD¹ and current studio practise recognises this Red Book channel as a 'bottleneck'. High-quality recordings are routinely made and edited using equipment whose performance potential is considerably higher than CD. Figure 1 illustrates this conceptually, while figure 2 illustrates how resolution (in this case indicated by word-size) typically varies through a quality audio chain.

Along the way, some interesting ideas have been proposed to try to maximise the human-auditory potential of CD. One idea is noise-shaping. Noise-shaping was first proposed by Michael Gerzon and Peter Craven in 1989 [11] and successfully embodied in Meridian's 618, 518 [12, 22] and also in Sony's Super Bit Mapping [3]. This technique has been used on maybe a few thousand titles – but these include some of the very finest sounding CDs available today. Other proposals were interesting, but didn't get off the ground – like subtractive dither and schemes to add bandwidth or channels to CDs [4, 14, 19,].

The author has felt strongly for some time, that we are on the threshold of the most fantastic opportunity in audio. It comes from two directions. First, psychoacoustic theory and audio engineering may have progressed to the point where we know how to define a recording system that can be truly transparent as far as the human listener is concerned. Second, we will soon see the evolution of a high-density audio format, related to DVD, that has, if it is used wisely, the data capacity to achieve this goal.

"coding high quality digital audio"

ACKNOWLEDGEMENTS

This article is based on papers the author has presented to the Audio Engineering Society. It inevitably, draws on the work of others. Many of its insights have resulted from discussions between members of the Technical Subcommittee of the Acoustic Renaissance for Audio (Tony Griffiths, Professor Malcolm Hawksford, David Meares and Bob Stuart) and their advisors (Peter Craven, Michael Gerzon, Hiro Negishi, Francis Rumsey and Chris Travis). The assistance of Peter Craven, Takeo Yamamoto, Mike Suzuki, Malcolm Law and Adrian Farmer has also been particularly valuable.

CONTACT ADDRESSES

Acoustic Renaissance for Audio, c/o Meridian Audio, Stonehill, Stukeley Meadows, Huntingdon PE18 6ED, England. Phone: +44 1480 52144 Fax: +44 1480 451587
 Email: ara@meridian-audio.com Web: <http://www.meridian-audio.com/ara>

Hörbereich: höchste Frequenzen?

IN-BAND NOISE SHAPING AND PRE-EMPHASIS

It is possible to exploit the frequency-dependent human hearing threshold by shaping the quantisation and dither so that the resulting noise floor is less audible.

Figure 19 shows how the Meridian 518 (an in-band shaper) can allow a 16-bit transmission channel to have a *subjective* noise floor more equivalent to a 20-bit 'simple' channel. If such a channel is to be useful, the resolution of the links in the chain before and after the noise-shaped channel must be adequate. In simple terms, this means mastering and playing back using well-designed converters offering at least 20-bit resolution.

It was the view of the ARA committee that noise shaping can be a linear process, and that it deserves serious consideration when distribution channels are to be matched to data-rate limitations.

FREQUENCY RANGE

The graphs to date have used the standard hearing threshold described in [20]. However, individuals can exhibit somewhat different thresholds [21 and 8]. The minimum audible field has a standard deviation of approximately 10dB.

Individuals are to be found whose thresholds are as low as -20dBspl at 4kHz. Similarly, although the high-frequency response cut-off rate is always rapid, certain people can detect 24kHz.

Abtastrate?

DO WE NEED MORE THAN 44.1KHZ?

The high-frequency region of figure 21 is shown in detail in figure 22. It can be seen that an average listener will find little to criticise in the in-band amplitude response of the DAC. To acute listeners, a 44.1kHz sample rate (even with the extremely narrow transition band shown) means a potential loss of extreme HF (between 20kHz and 22kHz). Raising the sampling rate to 48kHz does a lot to remedy this.

However, the significance of this has to be questioned. Although there is an area of intersection between the channel frequency responses and the hearing thresholds, this region is all above 100dBspl. The author knows of no program material that has any significant content above 20kHz and 100dBspl!

Numerous anecdotes suggest that a wider-frequency response 'sounds better'. It has often been suggested that a lower cut-off rate would give a more appropriate phase response, and that the in-band response ripple produced by the kind of linear-phase high-cut-off-rate filter illustrated in figure 22 (DAC) and figure 23 (ADC) can prove unexpectedly easy to detect. It is also frequently asserted that the slower rate of fall-off in HF response found in an analogue tape recorder accounts for a preferred sound quality.

It has also been suggested that the pre-ringing produced by the very steep linear-phase filters used so far for digital audio, can smear arrival-time detection and impact stereo imaging. This pre-ringing shows up in nearly all reviews of CD players. It can be significantly reduced by making the filter less steep (which we could do by raising the sample rate) or by not using a linear-phase characteristic.

The literature can contribute very little to this discussion. One well-performed set of experiments by Ohashi has, however, strongly indicated that certain program material may benefit from a system frequency response extending beyond 50kHz [17, 18].

Digitale Audioverarbeitung | WS 2000 | 18.205

Abtastrate?

PSYCHOACOUSTIC DATA ON HIGH-FREQUENCY HEARING

There is very little hard evidence to suggest that it is important to reproduce sounds above 25kHz. Instead there tends to be a general impression that a wider bandwidth can give rise to fewer in-band problems. However, there are a few points to raise before dismissing audible content above 20kHz as unimportant.

The frequency response of the outer and middle ear has a fast cut-off rate due to combined roll-off in the acoustics of the meata and in mechanical transmission. There also appears to be an auditory filter cut-off in the cochlea itself.

The cochlea operates 'top-down', so the first auditory filter is the highest in frequency. This filter centres on approximately 15kHz, and extrapolation from known data suggests that it should have a noise bandwidth of approximately 3kHz. Middle-ear transmission loss seems to prevent the cochlea from being excited efficiently above 20kHz.

Bone-conduction tests using ultrasonics have shown that supersonic excitation ends up in this first 'bin'. Any supersonic information arriving at above 15kHz therefore ends up here, and its energy will accumulate towards detection. It is possible that in some ears a stimulus of moderate intensity but of wide bandwidth may modify perception or detection in this band, so that the effective noise bandwidth could be wider than 3kHz.

The late Michael Gerzon surmised that any in-air content above 20-25kHz derived its significance from non-linearity in the hearing transmission, and that combinations of otherwise inaudible components could be detected through any resulting in-band intermodulation products.

There is a powerful caution against this. As far as the author knows, music spectra that have measured content above 20kHz always exhibit that content at such a low spl that it is unlikely that the (presumed) lower spl difference distortion products would be detectable and not masked by the main content.

Digitale Audioverarbeitung | WS 2000 | 18.205

Quantisierung?

24-BIT PCM CHANNELS

There is no convincing argument for using 24-bit data in a distribution format. Figure 7 clearly implies that the noise floor and resolution limit of a 24-bit channel will be 24dB greater than is necessary.

Why do it, then? One reason would be in order to convey more data for the subsequent DSP processes to work with. This reasoning is superficially correct. However, the author believes it to be unlikely that A/D converters that deliver 133dB analogue SNR will ever be made, and therefore a 24-bit channel would be kept busy conveying its own input noise! Furthermore, the majority of DSP systems and interfaces use a 24-bit word size. It is very, very difficult at present to guarantee transparency when performing non-trivial DSP operations on 24-bit data in a 24-bit processing environment. Obviously we could develop DSP processors capable of handling larger words, but why should we? Not only is the combination of well-handled, carefully delivered 20-bit data and a 24-bit processing environment good enough, but to deliver anything *more* is virtually to *guarantee* a higher risk of inadvertent truncation in the average replay chain.

A more pragmatic reason not to distribute 24-bit data is that it is virtually certain that the overwhelming majority of DVD players will not pass 24-bit data correctly. Even if they were to use 24-bit conversion, truncation is virtually guaranteed, whereas 20-bit data in the same pathway will pass virtually unscathed.

Digitale Audioverarbeitung | WS 2000 | 18.205

Quantisierungsfehler vs. Dithering

Single undithered truncations at the 16-bit level are regrettably all too common in practice. Not only do inadvertent truncations arise in the hardware filters of very many converters, but the editing and mastering processes often include level shifts, mixing events or DC filtering processes that have not been dithered correctly. There have therefore been reasonable grounds to criticise the sound of some digital recordings – even though (as laboured earlier) this particular defect can be avoided by combining good engineering with good practice.

Figure 14 represents the audible significance of a channel in which a correctly dithered quantisation (perhaps in a word-length reduction from 20 to 16 bits) is followed by a minor undithered process, in this case a 0.5dB attenuation. This figure shows how just one undithered process can degrade a correctly converted signal. Once again it is predicted that detection of a raised and granular noise floor is highly probable.

Figure 15 shows how this effect could operate in practice. The upper curve represents the audible significance of the same -90dBFS tone with all the errors introduced by an original 'correct' 16-bit quantisation followed by four undithered signal-processing operations. Four operations may seem like a lot, but this figure actually illustrates a common case in which everyday analogue-to-digital and digital-to-analogue converters are used. (As has already been mentioned, the decimation/oversampling filters in hardware converters are rarely dithered).

This curve may be taken as a baseline of current bad practice in CD recording/replay. It is put in historical context in figure 16, which includes the audible significance of the playback noise in a silent LP groove.

Digitale Audioverarbeitung | WS 2000 | 18.205

Audioqualität: Fazit

CONCLUSIONS

This article has reviewed the issues surrounding the transmission of high-resolution digital audio. It is suggested that a channel that attains audible transparency will be equivalent to a PCM channel that uses:

- 58kHz sampling rate, and
 - 14-bit representation with appropriate noise shaping, or
 - 20-bit representation in a flat noise floor, i.e. a 'rectangular' channel
- This conclusion has the following obvious implications:

- The CD channel with 44.1kHz 16-bit coding (even with noise shaping to extend the resolution) is inadequate
- Even 48kHz sampling is not quite high enough
- Sampling at 88.2kHz or 96kHz is too high, and therefore wasteful of data
- The use of sampling rates above 96kHz to convey a wider audio bandwidth cannot currently be justified

On the assumption that the industry will chose sample-rates based on 44.1kHz or 48kHz (i.e. 88.2kHz and 96kHz), we have looked at options for improving coding efficiency at these rates.

Noise shaping combined with a new pre-/de-emphasis characteristic for 96kHz (88.2kHz) applications can result in an effective addition of between 2 and 7 bits to the channel. In other words, at these sampling rates a 16-bit channel should be sufficient¹⁰.

Audioqualität: CD

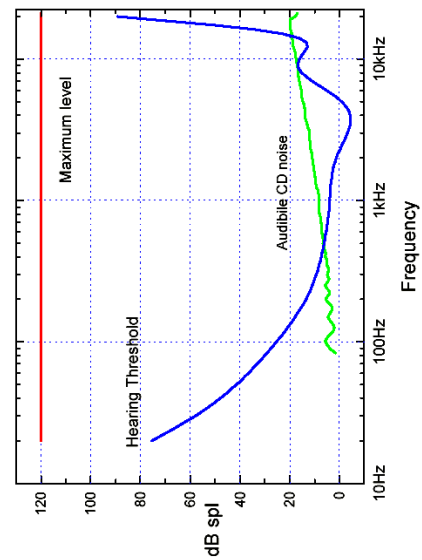


Figure 10. Dynamic range of CD.

Audioqualität: FM (UKW-Radio)

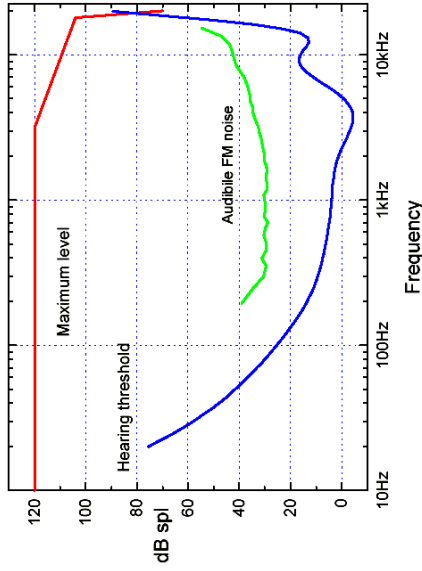


Figure 11. Dynamic range of FM.

Audioqualität: LP

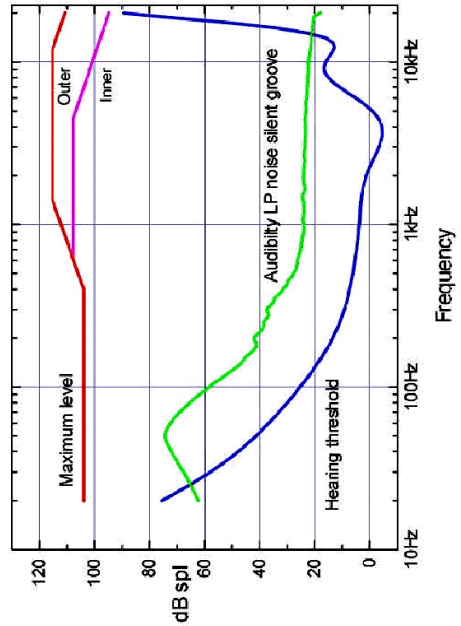


Figure 12. Dynamic range of LP.

Audioqualität: 16/18/20 bit

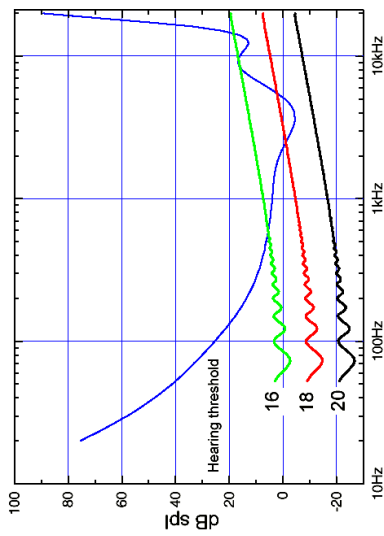


Figure 7. Audible significance of the noise created by a single white-spectrum TPDF-dithered quantisation in channels using 16, 18 and 20 bits. Audibility has been plotted against the average human hearing threshold assuming that a full-scale signal can attain 120dB spl at the listening position.

Audioqualität: 16/18/20 bit

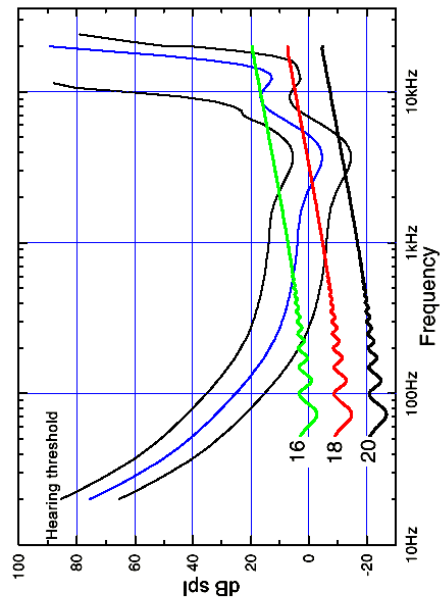


Figure 20. As figure 7, but showing how individual hearing thresholds can vary.

Audioqualität: Verzerrungen

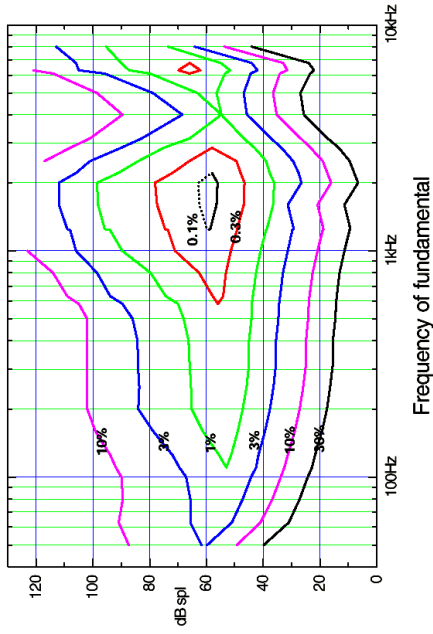


Figure 3. A contour map showing the existence regions for detecting the presence of an added second-harmonic tone. The spl is of the fundamental frequency. Inside a contour, 2nd-harmonic distortion of the marked percentage should be audible.

Audioqualität: Differenztöne

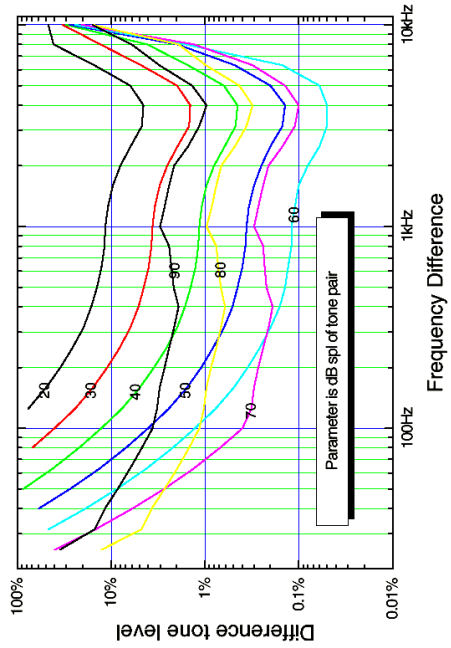


Figure 4. Illustrating predicted detectability of a difference-tone resulting from non-linear processing. See text. The parameter is spl of the combination.

Dither: Sinus ohne Dithering

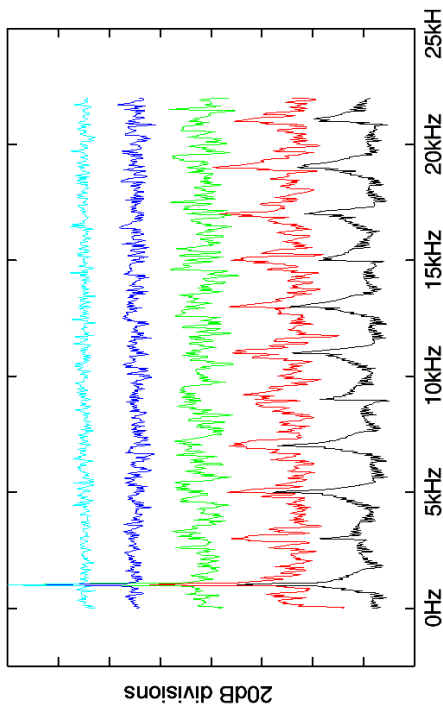


Figure 5. FFT analyses of an undithered 16-bit quantisation of a 1kHz tone at -20, -40, -60, -80 and at -90dBFS (top to bottom). Curves offset by 25dB.

Dither: Sinus mit Dithering

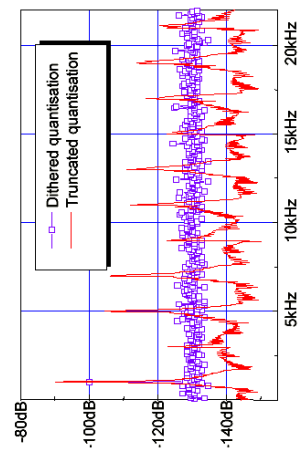


Figure 6. FFT measurements of the spectrum that results when a -90dBFS 1kHz tone is quantised to a 16-bit format, with and without correct (triangular probability distribution) dither.

Dither: ein Quantisierungsschritt

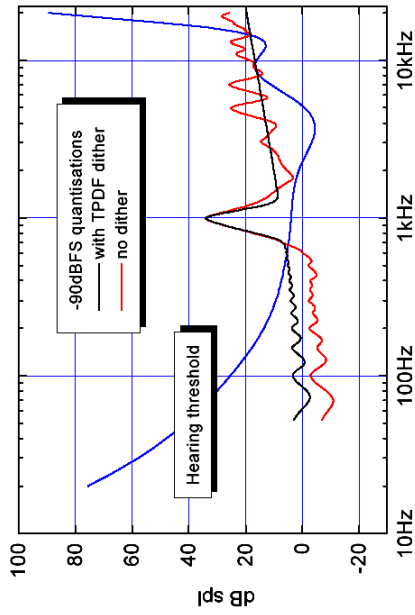


Figure 13. Audible significance of dithered and undithered 16-bit 44.1kHz sampling of a 1kHz -90dBFS (i.e. 30dBspl) tone. (0dBFS = 120dBspl.)

Dither: 1+4 Quantisierungsschritte

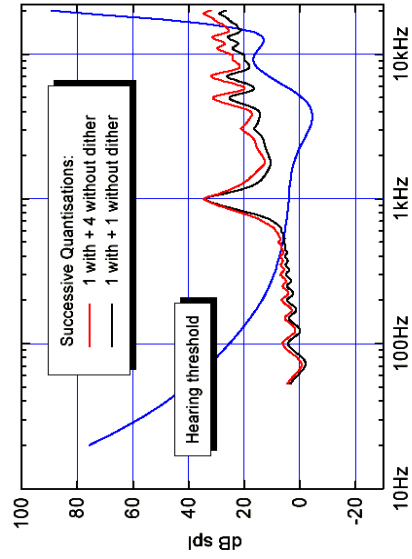


Figure 15. Audible significance of one (lower) and four (upper) successive undithered 16-bit 44.1kHz resamplings of a 1kHz -90dBFS (i.e. 30dBspl) tone on a signal already correctly quantised to 16 bits.

Noise-Shaping, 16 bit

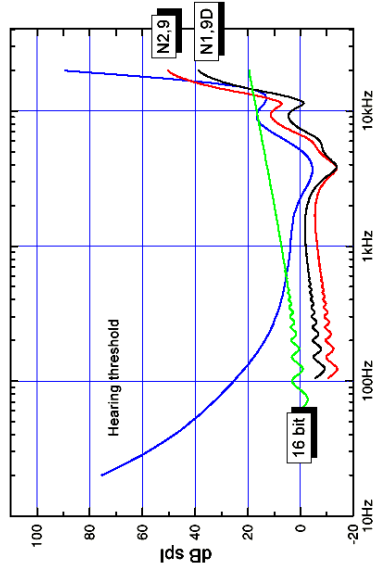


Figure 19. Audible significance of a simple 16-bit channel, with two examples from [24] of the audible significance of noise shaping in a 16-bit channel.

Audioqualität: CD/DAT Grenzfrequenzen

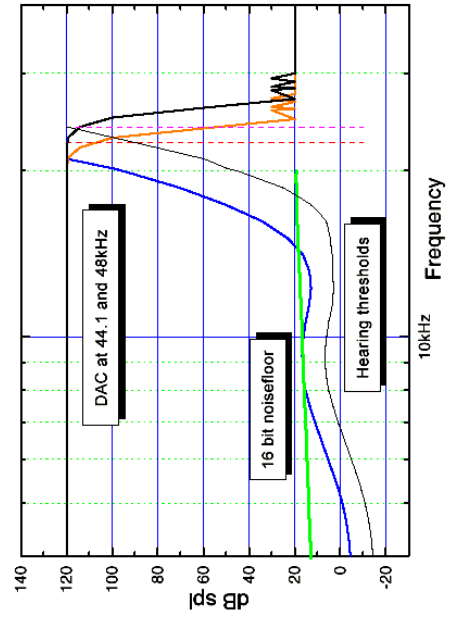


Figure 22. The high-frequency range of figure 21.

Dither: CD mit Quantisierung, vs. LP

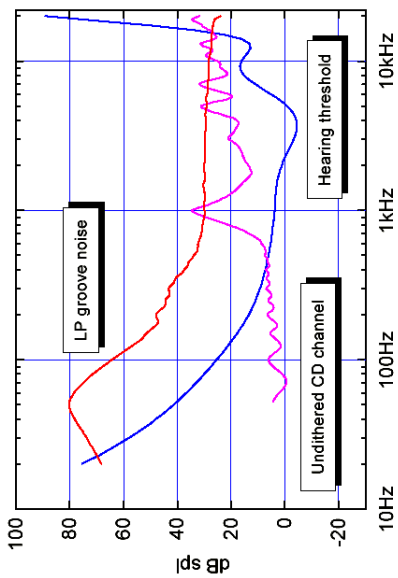


Figure 16. Audible significance of four (lower) successive undithered 16-bit 44.1kHz resamplings of a 1kHz -90dBFS (i.e. 30dB SPL) tone on a signal already correctly quantised to 16 bits, contrasted with the audible significance of noise floor measured on a silent LP groove.

Dither: Quantisierung, 20 bit

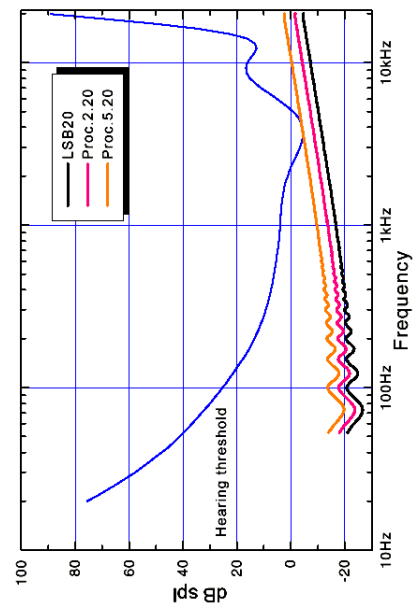


Figure 17. Audible significance of the noise created by 1, 2 and 5 successive TPDF-dithered quantisations in a 20-bit channel.

Audioqualität: 20 bit

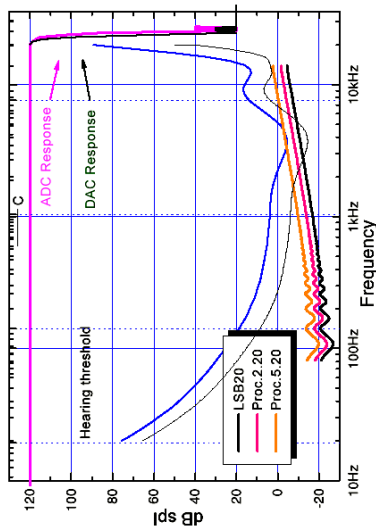


Figure 23. Useful operating region of a well-engineered 20-bit channel. The audible significance of noise created by 1, 2 and 5 successive quantisations is shown.

Audioqualität: "Shannon-Space"

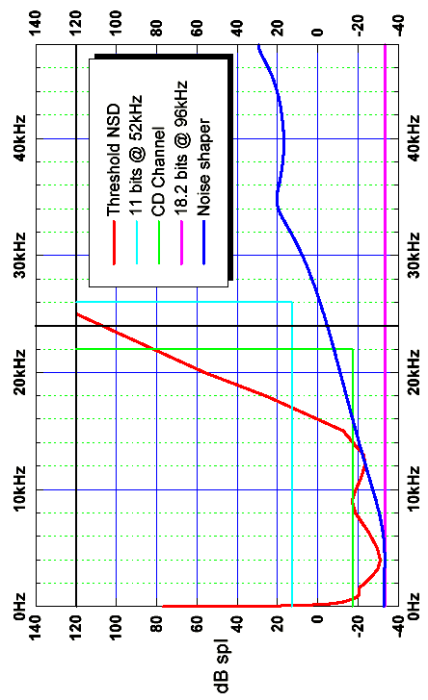


Figure 25. The 'Shannon space' for human hearing.

Audioqualität: 96 KHz, 66 KHz

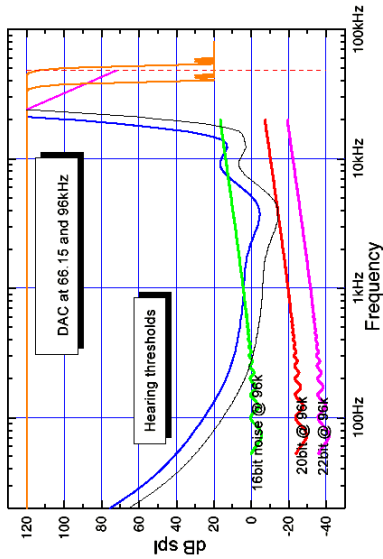


Figure 26. Useful operating regions of channels using 96kHz and 66.15kHz sampling. The figure shows that both rates allow for a near-audible HF region in which more gentle filtering could be used. The audible-significance channel-noise curves are given for 96kHz and for 16, 20 and 22-bit word lengths.

Audioqualität: 66 KHz, noise-shaping

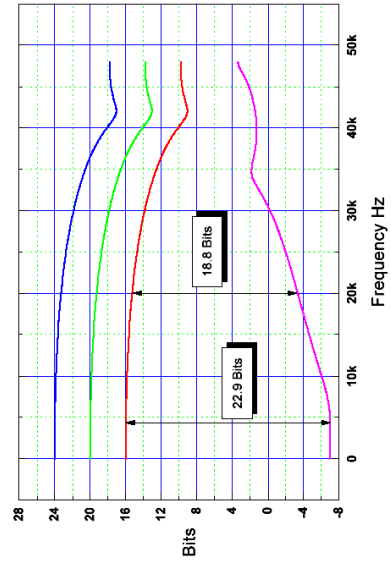


Figure 33. Output noise spectrum and headroom for a channel after the example 6th-order noise shaper has been combined with the proposed pre- and de-emphasis. The graph expresses the dynamic range in bits. The example illustrates a capacity of almost 23 bits at 4kHz for a 16-bit channel, i.e. a perceptual gain of 7 bits.

Audioverarbeitung: Systeme

Audio-Anforderungen an Rechensysteme?

- Grundoperationen
 - Quantisierung
 - Abtastrate
 - Algorithmen
 - weitere Anforderungen
 - Mikroprozessoren
 - SIMD-Befehlsätze
 - Signalprozessoren
 - ASICs
 - Analogrechner
- Addition, Multiplikation, Speicherung
> 16 bit, Gleitkomma
> 44 KHz
Rechenleistung, Speicherbedarf
Verschlüsselung, Mobilität, ...
Beispiel Memorystick
x86, PowerPC, ...
MMX, SSE, SSE2, ...
56K, SHARC, ...
EMU10K1
SSM-Chips

Audiosysteme: Literatur

- Zölzer: Digitale Audiosignalverarbeitung, Kap. 4
tech-www.informatik.uni-hamburg.de/lehre/pc-technologie/
developer.intel.com (x86 und MMX/ISSE Datenblätter)
developer.intel.com/tj (Intel technical journal)
www.motorola.com (56K, 96K Signalprozessoren)
www.analog.com (SHARC Signalprozessoren)
www.soundblaster.com (EMU10K1 und SoundFont Docs)

Audiosysteme: Anforderungen

Grundanforderungen:

- CD/DAT Qualität
 - höhere Genauigkeit
 - oder Gleitkomma
- 16 bit linear, 44.1 / 48 KHz
24 bit linear PCM
32 bit IEEE
- Beispiele: # MAC/s (48 KHz)
- 10 Kanal Mixer 480.000
 - 100-tap FIR-Filter 4.800.000
 - 1000-tap FIR-Filter 48.000.000
 - 10 Kanal Mixer, je 3 100-tap FIR 144.480.000
 - . . .
 - Pentium-200 (10 Takte/Mult.) 20.000.000
 - Pentium-III, 1 GHz, ISSE (max:) 4.000.000.000
- => Spezialhardware oder modern(st)e Prozessoren

Audiosysteme: Anforderungen

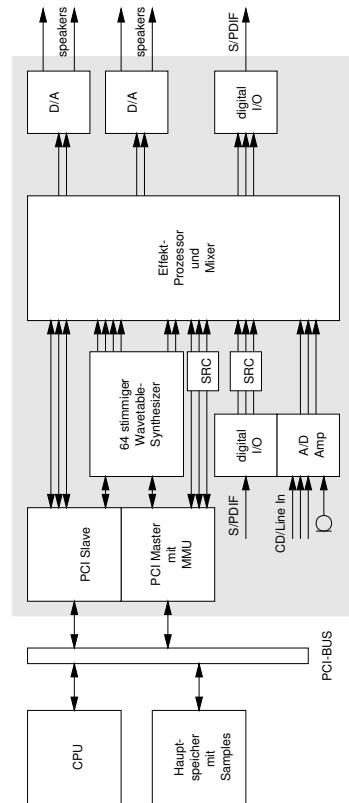
- Pentium-200 (10 Takte/Mult.) # MAC/s 20.000.000
 - Pentium-III, 1 GHz, ISSE (max:) 4.000.000.000
 - ältere PCs/MACs mit leistungsschwacher FPU für Audio/Multimedia ungeeignet
- aktuelle Prozessoren: (PC, Mac, Workstations)
- verbesserte FPU: geringe Latenz (typ. 3-5 Takte), Pipelining
 - SIMD-Erweiterungen: MMX, 3Dnow!, SSE, SSE2, AltiVec
 - zusätzlich: Moore's Law, höhere Taktraten
 - teilweise leistungsfähiger als Signalprozessoren - aber teurer
- => Verlagerung von Audiofunktionen in SW (Beispiel AC97)
 => PCs können viele HW-Geräte ersetzen
 => aber Marktbedeutung von Signalprozessoren wächst

Audiosysteme: Soundblaster Live

Beispiel Soundblaster-Live: [www.sblive.com, www.emu.com]

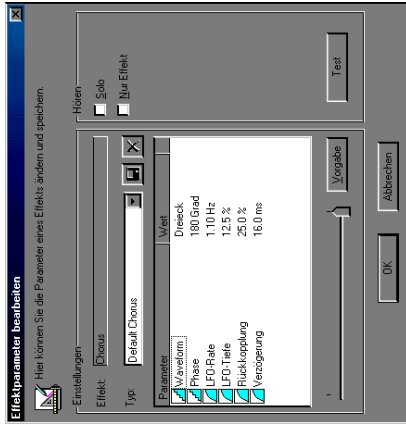
- state-of-the-art PC-Soundkarte
- Wiedergabe und Mixer mit 16 bit, 48 KHz
- 6 analoge Eingänge, 4 analoge Ausgänge
- S/PDIF Digitaleingang und -ausgang
- 64-stimmiger Synthesizer/Sampler (3 MIDI-Kanäle)
- reserviert PC-Hauptspeicher für Samples
- hochwertige Digitaleffekte
- Hardwareunterstützung für 3D-Audio
- unterstützt alle aktuellen Softwareschnittstellen
- ein zentrales ASIC: EMU10K1 Signalprozessor
- zusätzlich einige Analogbauelemente (Prof-Variante DM 1.000,00: bessere Wandler)
- ca. DM 150,00

Emu 10K1: Blockschaltbild



- PCI-Busmaster mit eigener MMU
- Samples im Hauptspeicher, per Treiber reservierte Bereiche
- vier analoge Ausgänge, digital S/PDIF inklusive 5.1

Emu 10K1: Effekte



- Vielzahl von digitalen Effekten, hier Beispiel Chorus
- Hall, Echo, Chorus, Phaser, Pitch-Shifter, . . .
- jeweils diverse Parameter editierbar

Emu 10K1: Interna

aktuell:

- Creative Labs unterstützt open-source Treiber für Linux
- opensource.creative.com

alle (?) internen Register in den Headerfiles dokumentiert

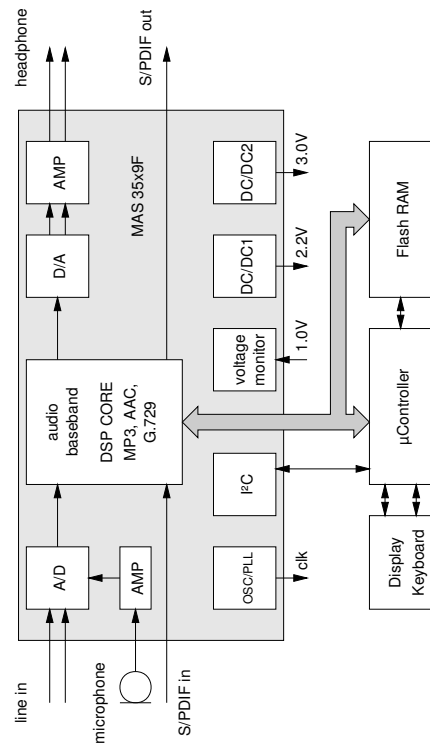
- Mixer
- Sampleratenumsetzung
- digitale I/O
- Synthesizer
- Digitaleffekte
- aber bisher nur ansatzweise implementiert (Mixer, S/PDIF out)

Audiosysteme: Micronas 39x9F

- 20-bit single-chip DSP
 - MPEG-1 Layer 1/2/3 Decoder
 - MPEG-2 AAC (low-bitrate) Decoder
 - Sprachcoders, Firmware-Download möglich
 - on-chip Stereo A/D und D/A
 - on-chip S/PDIF I/O
 - on-chip serielle / I2C Schnittstellen
 - on-chip DC-DC Wandler
 - on-chip Taktgenerator
- => 70 mW Stromverbrauch, 2.2 V
=> Batteriebetrieb, portable Geräte: MP3-Player, usw.

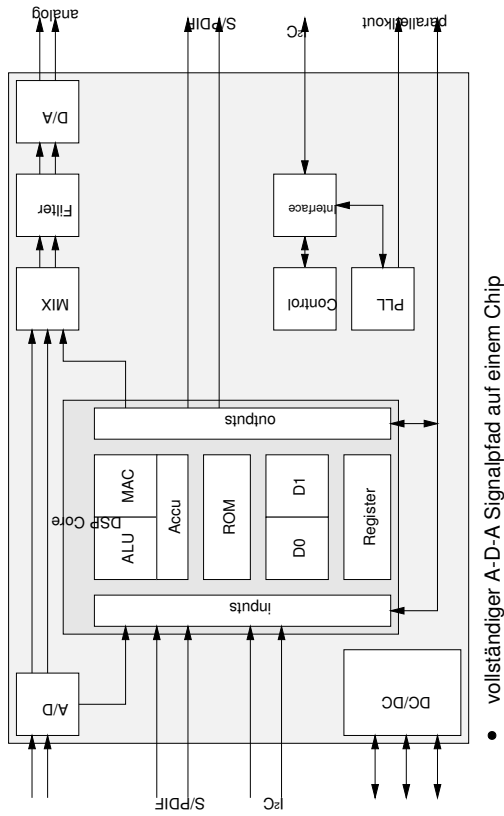


Micronas 39x9F: Applikation



- "portable audio player": 3 ICs plus Batterie / Tastatur / Display

Micronas 39x9F: Blockschaltbild



Micronas 39x9F: Features

- Datenregister 256 x 20 bit
 - Datenspeicher 4K x 20 bit
 - Programmspeicher 4K x 20 bit
- "high-level" Befehlssatz:
- Lautstärkeregelung
 - Filterung
 - MPEG-1 Layer 2/3 Dekodierung
 - Download zusätzlicher Programme
 - programmierbare DC/DC Konverter (Batteriebetrieb)
 - programmierbarer Taktgenerator
 - enorme Bedeutung der "analogen" Komponenten

Signalprozessoren

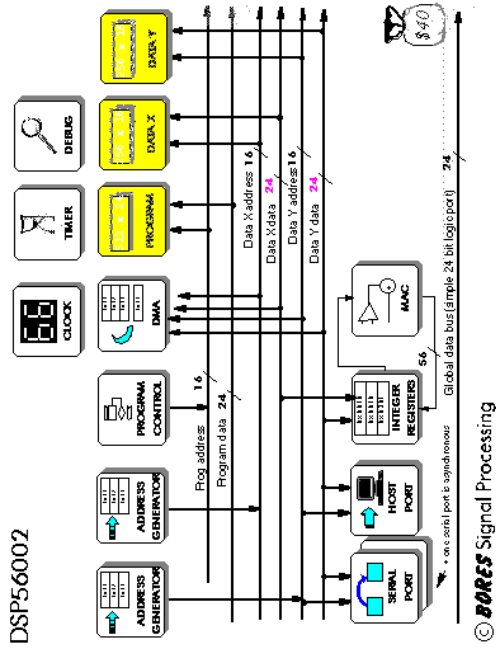
typische Algorithmen der Signalverarbeitung:

- MAC-Operationen
 - Programme mit vielen kurzen Schleifen
 - typische Adressierungsmuster (linear, modulo, FFT)
- darauf spezialisierte Hardware: "Signalprozessoren"
- ALU berechnet MAC-Operation in einem Takt
 - Akkumulator intern mit höherer Wortbreite
 - saturation Arithmetik
 - Harvard-Architektur mit separaten Bussen für Befehle/Daten
 - Busbandbreite ausreichend für single-cycle MAC
 - Adressberechnungseinheiten
 - Steuerwerk mit HW-Unterstützung für Schleifen und Interrupts

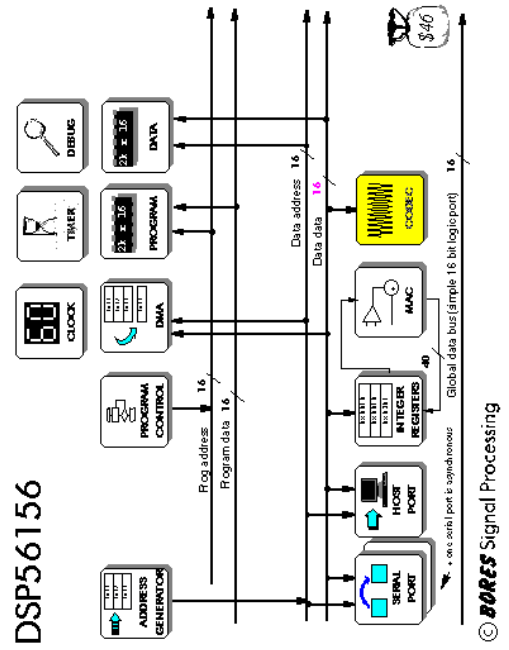
Signalprozessoren: Spektrum

- Festkomma / Gleitkomma
- Wortbreiten: 12 bit / 16 bit / 24 bit / 32 bit FP / extended . . .
- Adreßraum: 64 KB .. 4 GB extern
- on-chip Speicher: RAM / ROM (Tabellen) 1KB .. 1 MB
- Anzahl / Funktionen der ALU
- SIMD: mehrere ALUs
- Bandbreite der on-chip Busse
- Beispiele: Motorola 56K, 96K, Texas 32C0xx, AD SHARC
- teilweise extrem komplexe Parallelverarbeitung (TI 32C080)

Signalprozessoren: Motorola 56K

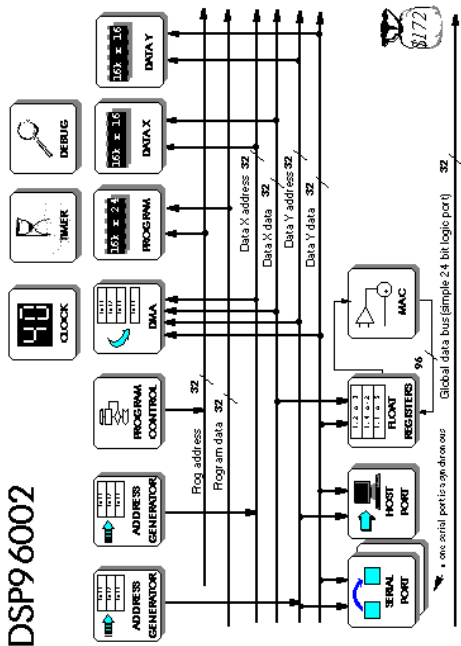


Signalprozessoren: Motorola 56K



Signalprozessoren: Motorola 96K

DSP96002



© **BORES** Signal Processing

SHARC

"Super Harvard Architecture"

- Signalprozessor-Familie, Analog Devices, www.analog.com
- Gleitkomma-Verarbeitung
- Datenformate 32 bit und 40 bit
- hohe Rechenleistung, 120 .. 600 MFLOPS peak
- Link-Ports für direkte Kommunikation
- erlaubt Parallelrechner mit sehr geringem Aufwand
- 32-bit Zahlenformat ideal für Audioverarbeitung
- oft in AV-Surround-Verstärkern eingesetzt
- Creamware Pulsar / Scope-System

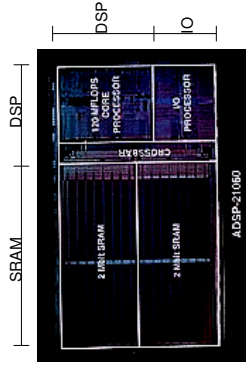
SHARC

ADSP-21060:

- 40 MHz, 120 MFLOPS peak
- 512 KB on-chip RAM
- Größenverhältnis RAM / Core (!)

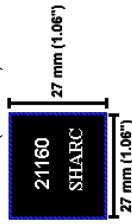
ADSP-21160:

- 100 MHz, 600 MFLOPS
- 500 MB/s external IO
- 600 MB/s link port IO
- 512 KB on-chip RAM
- 2 Watt



Layout ADSP-21060:

Platzbedarf (Platine):



SHARC

Aufbau des SHARC AD-21060:

- grobe Hardwarestruktur
 - Processing Unit mit Registern und drei Rechenwerken
 - Befehlssatz
 - Link-Ports und Parallelrechner
- siehe SHARC Datenblatt und Programmbeispiele



[Creamware Pulsar]

SIMD: Media processing

"Media processing" mit dem PC ?!

- steigende Anforderungen für Audio, Video, Image, 3D
- grosse Datenmengen
- aber oft mit geringer Genauigkeit (8 bit .. 16 bit, 32 bit FP)
- x86-FPU ausgereizt

=> Trick: vorhandene ALUs/Datenpfade für SIMD verwenden

Befehlssatzerweiterungen:

- MMX "multimedia extension" 1996
- 3DNow! 1998
- ISSE "internet SIMD streaming extension" 1999
- Altivec (PowerPC G4, Macintosh) 1999
- ISSE2 2000

SIMD: Flynn-Klassifikation

SISD

"single instruction, single data"

=> jeder klassische PC

SIMD

"single instruction, multiple data"

=> Feldrechner/Parallelrechner

=> z.B. Connection-Machine 2: 64K Prozessoren

=> eingeschränkt: MMX&Co: 2-8 fach parallel

MIMD

"multiple instruction, multiple data"

=> Multiprozessormaschinen

=> z.B. vierfach PentiumPro-Server

MISD

(-)

SIMD: Literatur

MMX: "The MMX technology page has been removed"

- developer.intel.com/drg/mmx/manuals/
- developer.intel.com/drg/mmx/appnotes/
- Linux "parallel-processing-HOWTO"
- IEEE Micro 8/96 S.42, c't 01/97 S.228ff

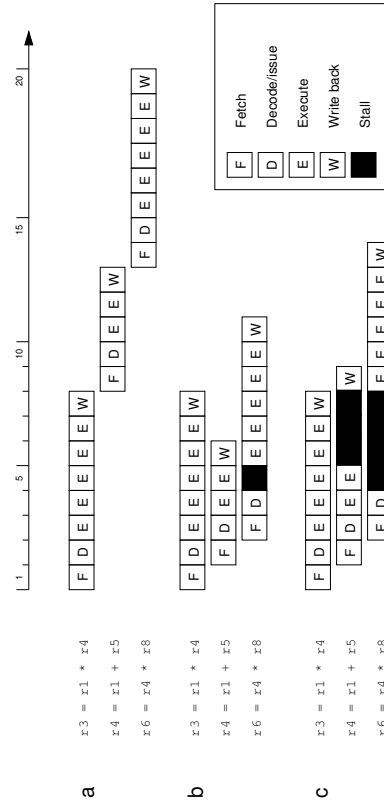
ISSE: Intel website:

- developer.intel.com/software/idap/resources/technical_collateral/pentiumiii/
- c't 04/00 S.314 (ISSE/3Dnow/Altivec)

3D Now! AMD website:

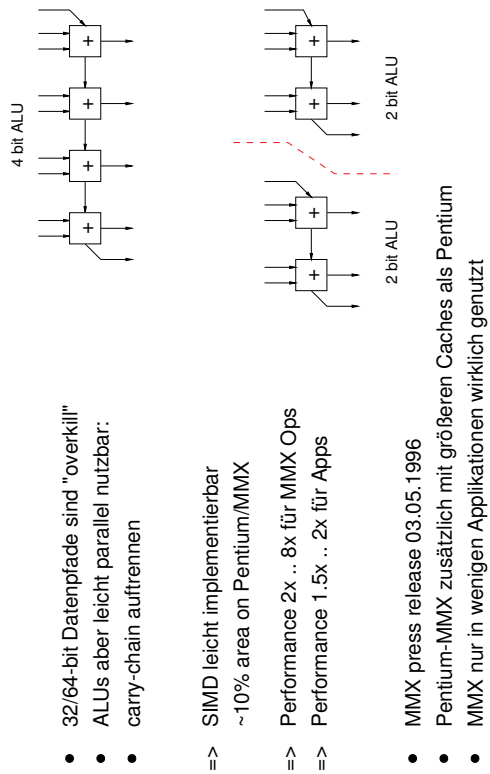
- www.amd.com/K6/K6docs/, www.amd.com/swdev/
- c't 15/98 S.186 ff
- IEEE Micro 3/4-99 S.37ff

Befehlspipeline: in order / out of order

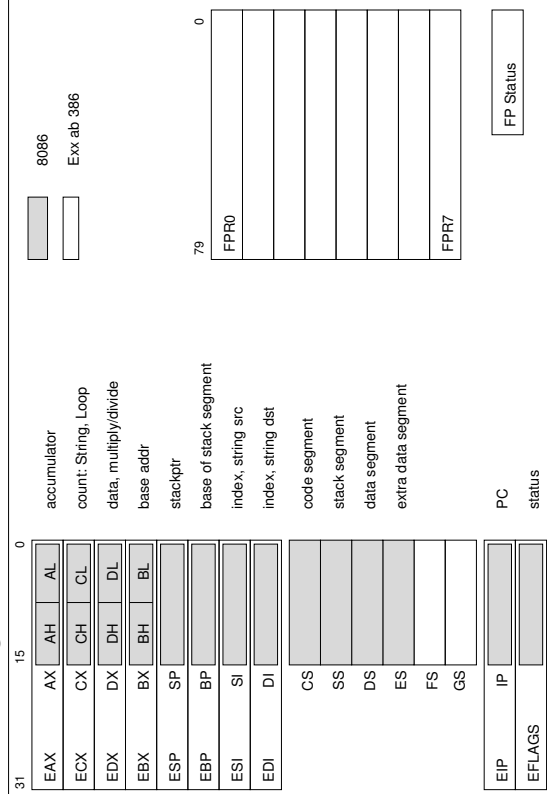


- a) serielle Befehlsbearbeitung
- b) pipeline, out-of-order completion
- c) in-order-completion

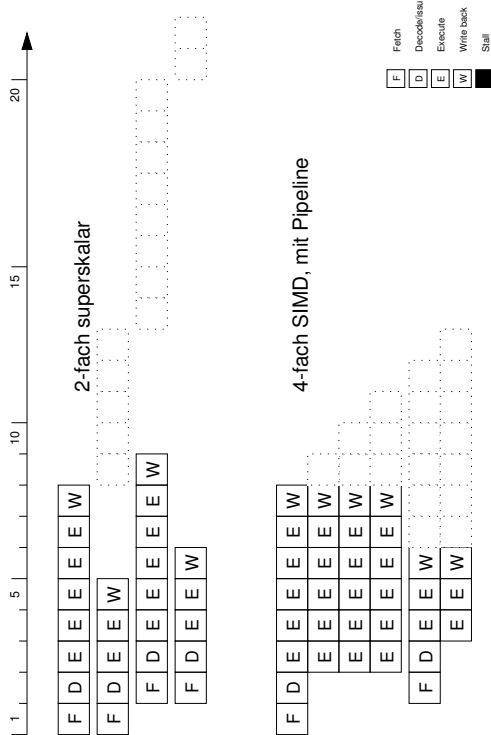
MMX: Grundidee



x86: Register



Superskalar, SIMD



Amdahl's Gesetz

"Speedup" durch Parallelisierung? [Gene Amdahl, 1967]

- System 1: berechnet Funktion X, zeitlicher Anteil 0<F<1
- System 2: Funktion X' ist schneller als X mit "speedup" SX: SX = Zeitbedarf(X) / Zeitbedarf(X')

Amdahl's Gesetz:
$$S_{\text{gesamt}} = \frac{1}{(1-F) + F/SX}$$

=> Optimierung lohnt nur für häufige Operationen !!

- => Beispiele:
- SX = 10, F = 0.1, Sgesamt = 1 / (0.9 + 0.01) = 1.09
 - SX = 2, F = 0.5, Sgesamt = 1 / (0.5 + 0.25) = 1.33
 - SX = 2, F = 0.9, Sgesamt = 1 / (0.1 + 0.45) = 1.82
 - SX = 1.1, F = 0.98, Sgesamt = 1 / (0.02 + 0.89) = 1.10

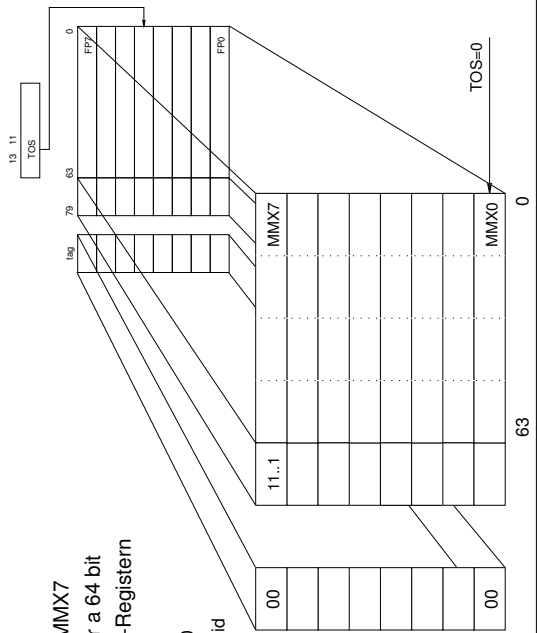
MMX: Entwurfsentscheidungen

Kompatibilität zu alten Betriebssystemen / Apps:

- keine neuen Register möglich
- keine neuen Exceptions
- bestehende Datenpfade nutzen
- möglichst wenig neue Opcodes
- alte Prozessoren und neue Software
- Test-Applikationen: (audio/image/MPEG-1/3D-Graphik/...)
- keine Tools
- optimierte Libraries verfügbar
- => FP-Register nutzen
- => Überlauf ignorieren
- => saturation arithmetic
- => 64 bit
- => Code doppelt
- => MMX DLLs
- => 16 bit dominiert
- => Assembler

MMX: Register

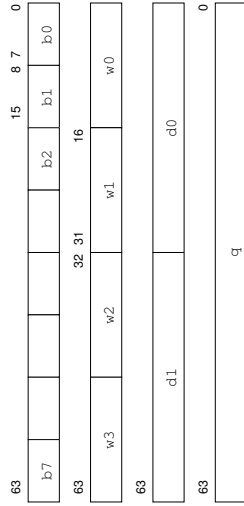
- MMX0 .. MMX7
- 8 Register a 64 bit
- in den FP-Registern
- FP NaN
- FP TOS = 0
- tag 00 = valid



MMX: Datenformate

64-bit Register, 4 Datentypen:

- packed byte *8 / packed word *4 / packed doubleword *2 / quadword
- Zugriff abhängig vom Befehl



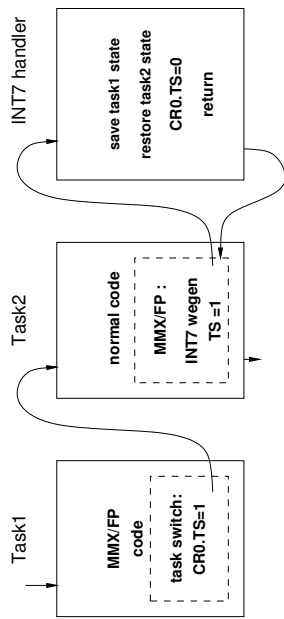
MMX: Befehlssatz

MMX (FSAV / FRESTOR)	Description
MOVQ mm1, mm2/mem32	clear MMX state (handle FP regs)
MOVQ mm1, mm2/mem64	move 32 bit data
PACKSSWB mm1, mm2/mem64	move 64 bit data
PUNPCKH mm1, mm2/mem64	pack 8*16 into 8*8 signed saturate
PACKSSDW mm1, mm2/mem64	fancy unpacking (see below)
PAND mm1, mm2/mem64	pack 4*32 into 4*16 signed saturate
PCMPQB mm1, mm2/mem64	mm1 AND mm2/mem64 / auch OR/XOR/NAND
PADDB mm1, mm2/mem64	8*a=b, create bit mask / auch GT
PSUBD mm1, mm2/mem64	8*add 8 bit data
PSUBQ mm1, mm2/mem64	2*sub 32 bit data / signed wrap
PSUBSD mm1, mm2/mem64	2*sub 32 bit data / unsigned saturate
PSLL mm1, mm2/mem64/imm8	shift left mm1 / auch PSRA/PSRL
PMULL/HW mm1, mm2/mem64	4*mul 16*16 store low/high 16 bits
PMADDWD mm1, mm2/mem64	MAC 4*16 -> 2*32

insgesamt 57 Befehle (Varianten B/W/D S/US)

MMX: Multitasking ...

Interaktion mit Betriebssystem / Taskwechsel:



- FP-Register nur bei Bedarf sichern
- vorhandene FP INT7 Routine funktioniert auch für MMX
- keine Anpassung des Betriebssystems notwendig

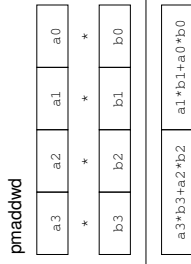
MMX: "Saturation Arithmetic"

was soll bei einem Überlauf passieren?

- **wrap-around**
... 125, 126, 127, -128, -127, ...
- **saturation**
... 125, 126, 127, 127, 127, ...
- **Zahlenkreis "aufgeschnitten"**
- gut für DSP-Anwendungen

MMX: "packed multiply add word"

für Skalarprodukte:



```

vector_x_matrix_4x4 ( MMX64* v, MMX64* m ) {
    MMX64 v0101, v2323, t0, t1, t2, t3;

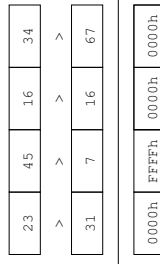
    v0101 = punpckldq( v, v ); // unpack v0/v1
    v2323 = punpckhdq( v, v ); // unpack v2/v3

    t0 = pmaddwd( v0101, m[0] ); // v0|v1 * first 2 rows
    t1 = pmaddwd( v2323, m[1] ); // v2|v3 * first 2 rows
    t2 = pmaddwd( v0101, m[2] ); // v0|v1 * last 2 rows
    t3 = pmaddwd( v2323, m[3] ); // v2|v3 * last 2 rows

    t0 = paddq( t0, t1 ); // add
    t2 = paddq( t2, t3 ); // //
    v = packssdw( t0, t2 ); // pack 32->16, saturate
}
    
```

MMX: "packed compare"

Vergleiche / Sprungbefehle:

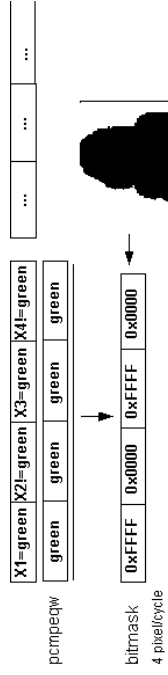


- schlecht parallelisierbar
 - Pipeline-Abhängigkeiten
- => keine Sprungbefehle in MMX
=> compare-Operationen setzen Bitmasken
- Bitmasken für logische Ops verwendbar
 - Beispiel: chroma-keying

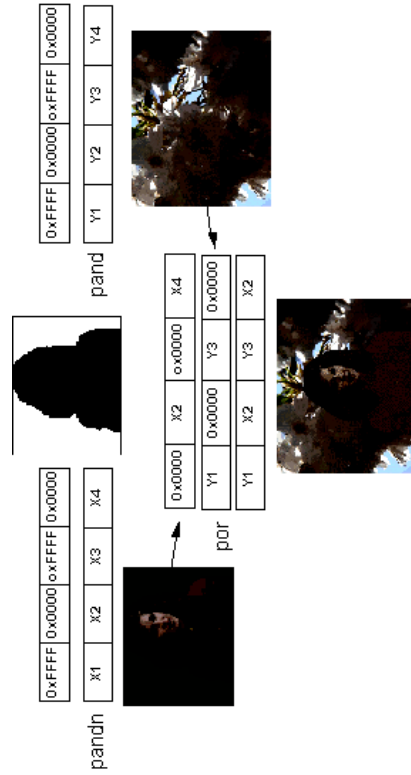
MMX: Chroma Keying (1)

"Weiterbericht":

- MMX berechnet 4 Pixel / Takt
- keine Branch-Befehle
- Schritt 1: Maske erstellen (high-color: 16 bit/pixel)

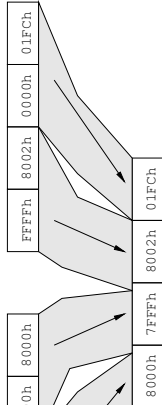


MMX: Chroma Keying (2)

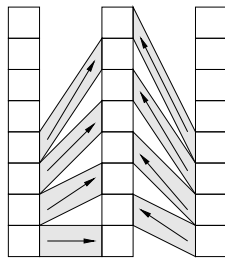


MMX: packssdw / punpckhbw

packssdw: pack with saturation 32 -> 16 signed data:



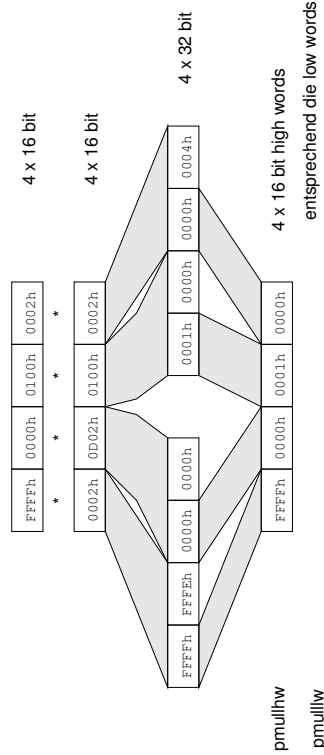
punpckhbw:



punpcklbw: lower 32 bits

MMX: pmullw / pmullhw

pmullhw: multiply 4 words, write low/high byte of results:



mit Packbefehlen kombinieren, wenn 32-bit Resultate gewünscht

MMX: Zufallszahlen

```

x(t) = (x(t-1) * 47989) & 0xFFFF;

QuadWord DithMultVal = 0x4f314f314f314f31;
QuadWord DithRegInit = 0x4f31994d2379bb75;

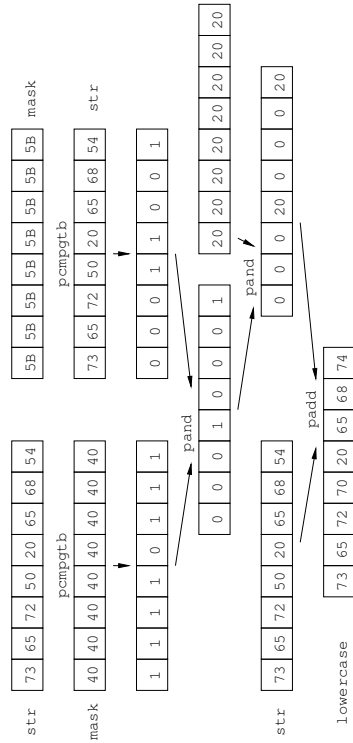
Init:
MOVQ    mm0, DithRegInit;

Loop:
PMULLW mm0, DithMultVal // x(t) -> x(t+1)
MOVQ    [result64], mm0 // 3 clocks
        // 1 clocks

• PMULLW latency 3, throughput 1 (on Pentium)
• bis zu vier Zufallszahlen pro Takt (UV pipelines genutzt)
    
```

MMX: toLowerCase()

String lower-to-upper-case conversion:



(aber Probleme mit Umlauten...) [aus Intel MMX appnote]

3Dnow! Motivation

- stark wachsende Bedeutung von 3D-Spielen
 - 32-bit Gleitkommaoperationen nötig für Geometrie-Transformationen
 - FPU im AMD K6 vergleichsweise langsam
 - MMX unterstützt nur Integer-Datentypen
- => SIMD-Befehle für 32-bit float Datentypen
- schnelle Add/Mult/MAC/Sqrt-Befehle
 - muß ohne OS-Unterstützung nutzbar sein
 - MMX-Register verwenden
 - MMX zwei-Operanden Adressierung
 - je zwei float-Datenwerte pro MMX-Register
- => 3Dnow! Spezifikation (vergleiche Motorola AltiVec / Intel ISSE)

3Dnow! Entscheidungen

SIMD-Befehle für 32-bit float Datentypen:

- MMX-Register verwenden, zwei Datenwerte pro Register
- zwei-Adress-Befehle
- keine Status-Flags, keine Exceptions
- MMX-Befehle nutzbar (logische, Vergleiche, ...)
- belegt nur einen einzigen x86 Opcode (0F0F ... subopcode)

möglichst wenig Chipfläche:

- keine Unterstützung für NaN/INF/...
- nur round-to-nearest-even Modus, +-1LSB
- Saturation-Arithmetik statt Überlauf
- Approximation für Division und Quadratwurzel

3Dnow! Prefetch

Speicherzugriffe in Multimedia-Applikationen:

- reguläre Speicherzugriffsmuster
 - ungewöhnliche Lokalität
 - viele Daten werden (pro Frame) nur einmal benötigt
 - aber regelmässig (in jedem Frame)
 - Performance stark von optimaler Cache-Ausnutzung abhängig
- => prefetch-Befehl
- quasi normaler Ladebefehl, aber ohne Zielregister
 - gewünschte Daten werden in L1/L2-Cache geladen
 - löst keine Exceptions / Page Faults aus
- => "memory streaming"
- => auch für andere Anwendung gut nutzbar (etwa Numerik)

PC-Technologie | SS 2000 | 18.215

3Dnow! Division / Quadratwurzel

- Rechenwerk für Division / Sqrt ist sehr aufwendig
 - möglichst wenig Chipfläche für 3Dnow!
 - teilweise nur geringe Genauigkeit benötigt
 - etwa Shading/Beleuchtungsberechnung für 3D-Graphik
- => Division und Quadratwurzel per Approximation
- erster Befehl liefert 14/15 bit Approximation
 - aus Lookup-Table und Interpolation
 - mit vollem Takt
 - zusätzliche Befehle für Newton-Iteration
 - quadratische Konvergenz: zwei Iterationsschritte für volle Genauigkeit
 - wenig Hardwareaufwand
 - voll in Pipeline integriert, maximaler Durchsatz

PC-Technologie | SS 2000 | 18.215

3D Now! Apfelmännchen

```
Function IterPasD
(I,R :Double; Grenze, Tiefe :Paratyp) :Paratyp;
var A,B,C:double;
Begin
Count:= 0;
A:=0; B:=0;
Repeat
C:= SQR(A) - SQR(B) + R;
B:= 2*A*B + I;
A:= C;
INC (Count);
Until (abs (A) >Grenze) or (Abs (B) > Grenze)
or (Count=Tiefe);
IterpasD:=Count;
End;
```

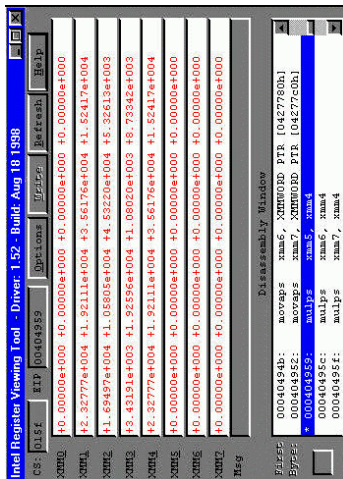
PC-Technologie | SS 2000 | 18.215

3D Now! Apfelmännchen

```
; Quadriere (A + jB)**2 = A**2 - B**2 + j 2*A*B
; Entry MM0 ; A | B
; MM1 ; 1 | -1
; MM2 ; R | I
;
; Loop:
MOVQ MM3,MM0 ; MM3=A | B
MOVQ MM4,MM0 ; oh weh
PSLLQ MM3,32 ; das Vertauschen ist
PSRLQ MM4,32 ; sehr mühsam ...
POR MM3,MM4 ; MM3=B | A
PFMUL MM3,MM0 ; MM3= A*B | A*B
PFMUL MM0,MM0 ; MM0= A**2 | B**2
PFMUL MM0,MM1 ; MM0= A**2 | -B**2
PFACC MM0,MM3 ; MM0= A**2 - B**2 | A*B+A*B
PFADD MM0,MM2 ; MM0= A**2 - B**2 + R | 2*A*B+I
; MM0= A(n+1) | B(n+1)
PEZID MM4,MM0 ; IA = INT(A) | IB = Int(B)
MOVQ IA,MM4
... ; Sieh nach, ob A oder B > GRENZE ist
dec CX ; iteration counter
jnz loop
```

PC-Technologie | SS 2000 | 18.215

ISSE: Register Viewing Tool



Softwareentwicklung für MMX / ISSE / 3DNow:

- nur rudimentäre Compiler- und Tool-Unterstützung
- oft handoptimierter Assembler wg. bester Performance

ISSE: Programmierung

Intel VTune Performance Enhancement Environment:

- optimierender Compiler mit ISSE-Unterstützung:
- Intrinsics
- Vector Class Library
- Vectorization
- Intel Performance Library Suite
- C-Funktionen, Compiler inlining
- Klassen, inlining durch Compiler optimierender Compiler
- erfordert 16-Byte Alignment aller Datentypen
- umfangreiche Profiling-Tools
- sehr teuer

ISSE: Programmierung mit "Intrinsics"

```
float xa[SIZE], xb[SIZE], xc[SIZE];
float q;

void do_c_triad() {
    for( int j=0; j < SIZE; j++ ) {
        xa[j] = xb[j] + q*xc[j];
    }
}

#define VECTOR_SIZE 4
__declspec(align(16)) float xa[SIZE], xb[SIZE], xc[SIZE];
float q;

void do_intrin_triad() {
    __m128 tmp0, tmp1;

    tmp1 = _mm_set_ps1(q);
    for( int j=0; j < SIZE; j+= VECTOR_SIZE ) {
        tmp0 = _mm_madd_ps( *( (__m128 *) &xc[j]), tmp1 );
        *( (__m128 *) &xa[j] ) = _mm_add_ps( tmp0, *( (__m128 *) &xb[j] );
    }
}
```

ISSE-Programmierung mit "Intrinsics" und VTUNE:

```
struct
{
    float X, Y, Z; // A, B, C;
} AoS_data[1000];
```

- Daten lokal
- Anordnung schlecht für SIMD

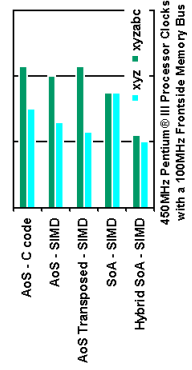
Structure of Arrays:

```
{
    float X[1000], Y[1000], Z[1000];
} SoA_data;
```

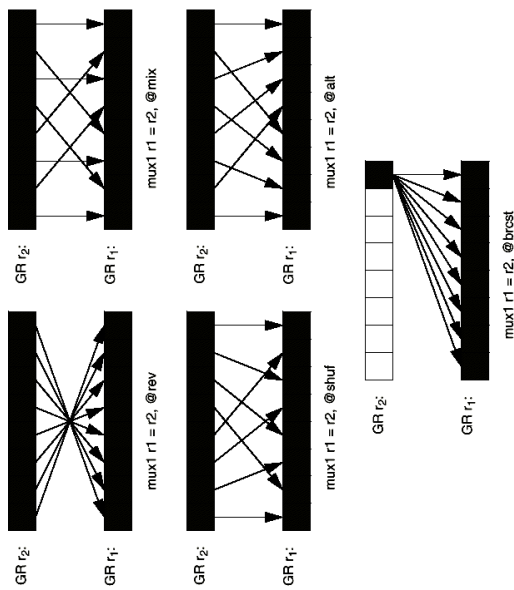
- Anordnung optimal für SIMD
- aber im Speicher "verstreut"

=> Hybrid SoA - SIMD

```
struct
{
    float X[8], Y[8], Z[8];
} Hybrid_data[125];
```



ISSE2: mux1-Befehl (IA64)



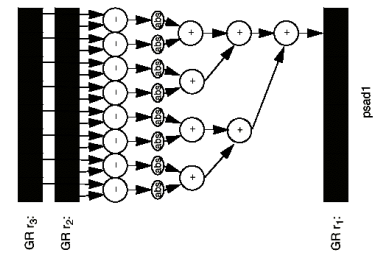
ISSE2: psad1-Befehl (IA64)

Parallel Sum of Absolute Difference

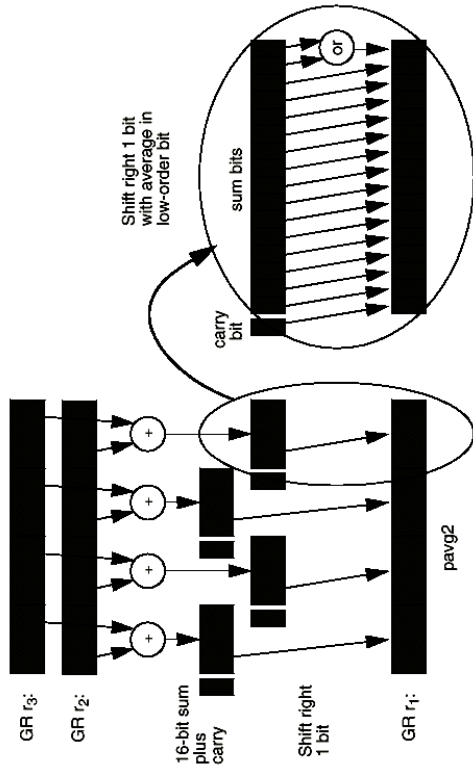
Format: (gp) psad1 r1 = r2, r3

Description: The unsigned 8-bit elements of GR r2 are subtracted from the unsigned 8-bit elements of GR r3. The absolute value of each difference is accumulated across the elements and placed in GR r1.

Figure 7-36. Parallel Sum of Absolute Difference Example



ISSE2: pavg2-Befehl (IA64)



ISSE: FIR-Filter

Nutzen von MMX/ISSE für Filter?

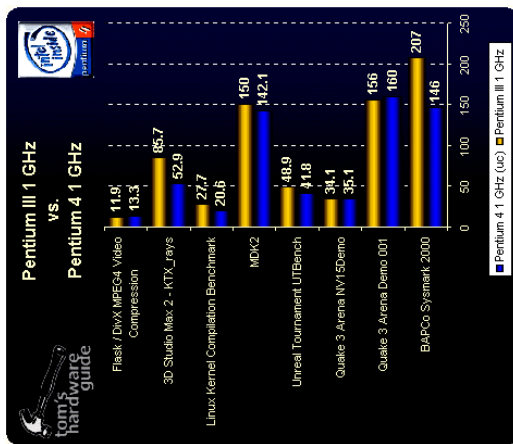
- MMX und ISSE für 16-bit Integer
- ISSE für 32-bit Gleitkommawerte
- maximal vierfache Leistung gegenüber skalarem Code

aber:

- erfordert Operanden-Alignment (16-Byte Grenzen)
- z.B. durch Duplizierung der Daten/Koeffizienten-Arrays
- Arraygrößen Vielfache von 4
- Multiplikation parallel, aber Akkumulation schwierig

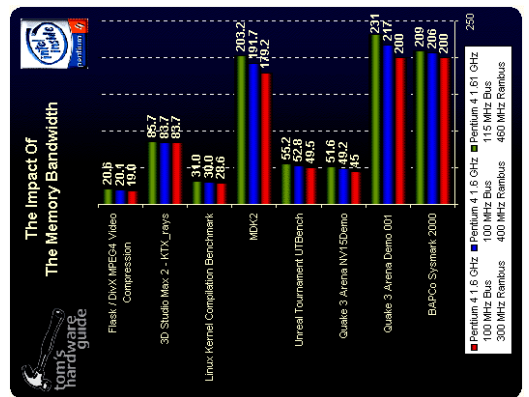
=> siehe Intel Appnote
"32-bit FP FIR Filter implemented using SSE"

Benchmarks: Pentium IV vs. Pentium III



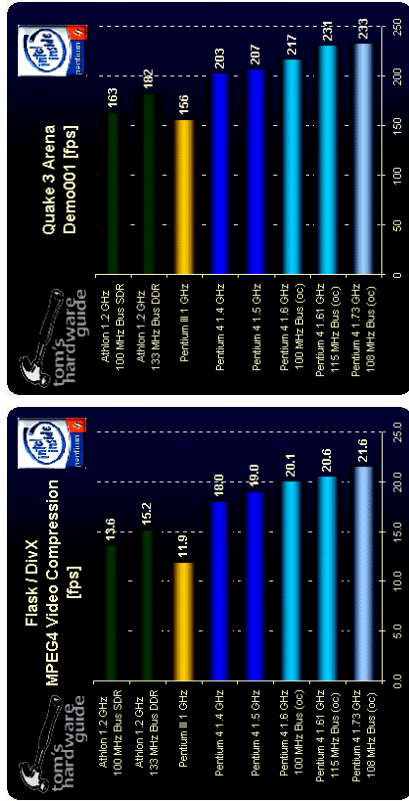
Digitale Audioverarbeitung | WS 2000 | 18.205

Benchmarks: Pentium IV, Speicher



Digitale Audioverarbeitung | WS 2000 | 18.205

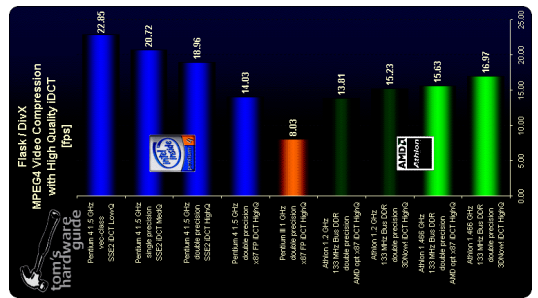
Benchmarks: DivX / Quake



(DivX mit MMX DCT-Code)

Digitale Audioverarbeitung | WS 2000 | 18.205

Benchmarks: DivX mit 3Dnow/ISSE2



Verwirrspiel im DivX-Benchmark:

- uneinheitliche Pentium-4 Ergebnisse
- sehr gute Werte mit MMX
- sehr schlechte Werte mit x86 FPU
- neu kompilierter Code:
- besser auf P3, P4, Athlon
- neu kompilierter Code mit ISSE2
- neu kompilierter Code mit 3Dnow!
- Graphik zeigt die jeweils besten Werte
- ISSE2 sehr leistungsfähig
- Speicherinterface wichtig (Rambus/DDR)

Digitale Audioverarbeitung | WS 2000 | 18.205

CD, DVD: Agenda

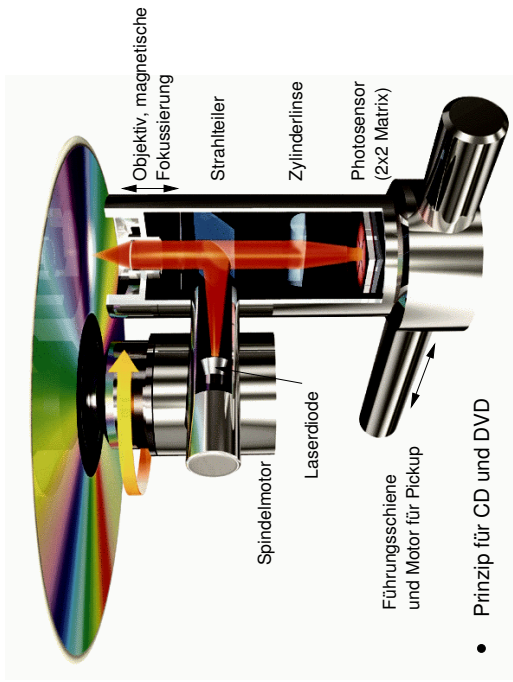
- Grundlagen der CD-Technik
- CD-ROM
- CD-R, CD-RW
- DVD und Video
- DVD-R, DVD-RAM
- ISO-9660 Dateisystem
- UDF / Packet-Writing



CD/DVD: Literatur

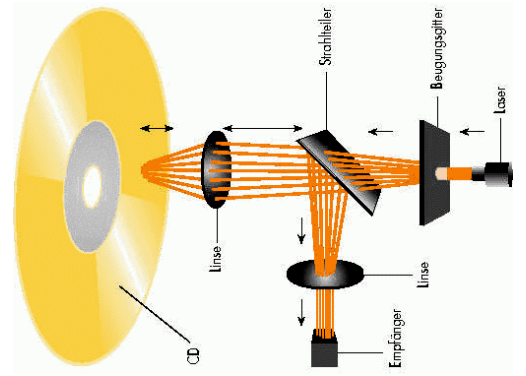
- www.discronics.co.uk/
 - www.fadden.com/cdrfaq
 - www.dvddemystified.com/dvdfaq.html
 - www.ping.be/~pin11466/formtx.html
 - www.unik.no/~robert/hifi/cdv/
 - www.phoenix.com/techs/specs.html
- diverse c't Artikel:
- 02/93 178ff CD "color books" und Dateiformate
 - DVD: 23/99 S.100ff
- diverse Standards, insbesondere ECMA-xxx (frei verfügbar), www.ecma.ch
- ECMA-119 ISO-9660 Dateisystem
 - ECMA-267 DVD-ROM Spezifikation
 - SCSI-3 MMC SCSI MultiMedia Command Set

CD: Aufbau eines Players



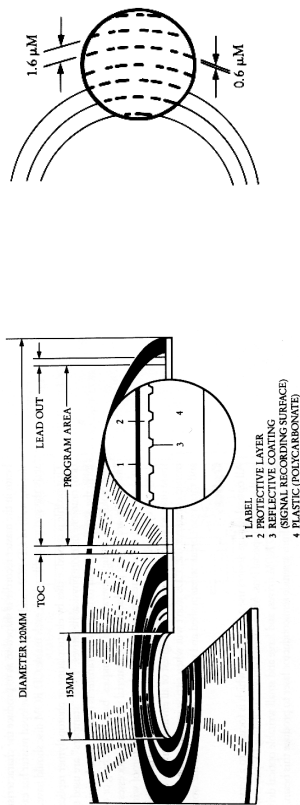
- Prinzip für CD und DVD

CD: Multibeam-Technik



- Strahlteiler
- komplexer Empfänger mit mehreren Photodioden
- liest mehrere Spuren gleichzeitig
- statt höherer Drehzahl

CD: Prinzip



- Polycarbonatträger, 12cm Durchmesser
- eingeprägte Vertiefungen ("pits") bilden die Daten
- spiralförmige Datenspur, 1.6µm Abstand, ca. 16000 Windungen
- Fertigungsmängel fest eingeplant => leistungsfähige Fehlerkorrektur

[CD-ROM - The new Papyrus]

CD: Reflexion

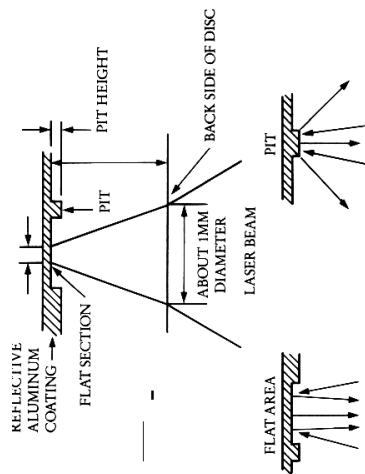
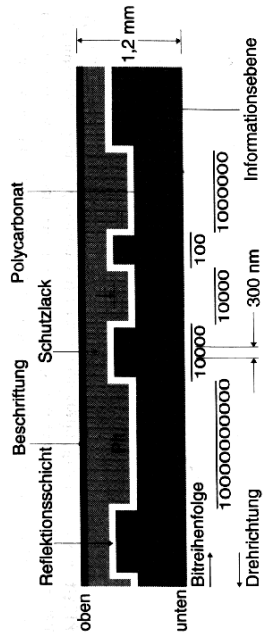


Figure 4. Relation between pits and photodetector output.

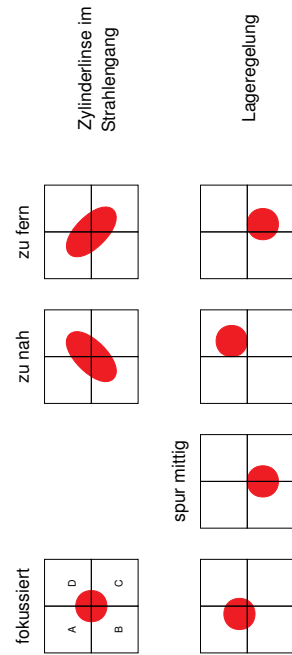
- Lands reflektieren das Laserlicht
- Pits streuen das Laserlicht

CD: Schichtaufbau



- Polycarbonatschicht ~ 1.2 mm
- Größe der Pits / Lands ~ 1.0 µm
- Interpretation: Land = 0, Pit = 1, Wechsel Land/Pit = 1
- Achtung: Kratzer oben zerstören die Daten

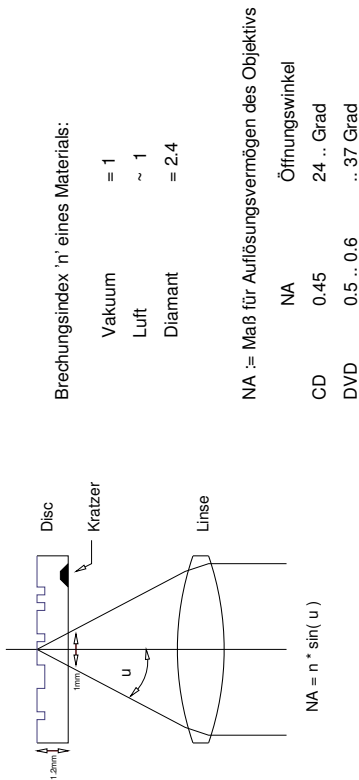
CD: Fokussierung, Spurregelung, ...



Sensorfeld mit 2x2 Photodioden zur Regelung:

- Fokussierung: aus Differenzsignal (A+C) - (B+D)
- Spurregelung: aus Differenzsignal (A+B) - (C+D)
- Nutzsignal: Land/Pit-Übergänge aus Summensignal
- Drehzahl: aus Taktfrequenz des Nutzsignals

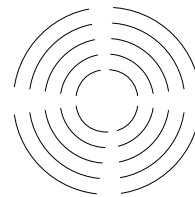
CD: numerische Apertur



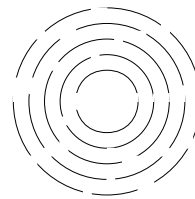
- Auflösungsvermögen wie bei Mikroskopen (!)
- asphärische Linsen notwendig
- Kratzer/Staub auf der Oberfläche stören kaum

CD: CAV vs. CLV

Constant Angular Velocity
(Floppy, aktuelle CD-ROM Laufwerke)



Constant Linear Velocity
(Audio/Video CD, DVD)



- Audiodaten: konstante Datenrate sinnvoll: CLV
- Drehzahl: innen hoch, außen langsam
- Angabe "48x"-Laufwerk: 48x Datenrate (CLV) der Audio-CD
- CAV erlaubt Spurwechsel ohne Drehzahländerung
- aktuelle CD-ROMs: CAV soweit per Daten/Fehlerrate möglich

CD: EFM

Eight-to-Fourteen Modulation:

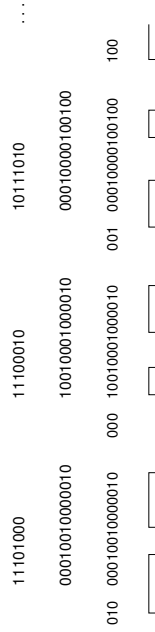
- selbsttaktende Aufzeichnung, NRZI
- minimal 2 Nullen, maximal 11 Nullen zwischen Einsen

data bits	channel bits
0000 0000	010010001000000
0000 0001	100001000000000
0000 0010	100100001000000
0000 0011	100010000100000
0000 0100	010000100000000
0000 0101	000001000100000
...	... via lookup table

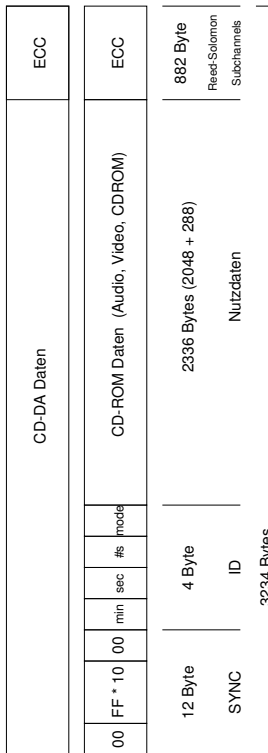
- zusätzlich 3 "Merge-Bits" zwischen zwei Codeworten einfügen
- eigentlich 8-17 Modulation
- DVD verwendet verbessertes 8-16 Verfahren

CD: Kodierung der Daten:

- Nutzdaten
- Nutzdaten in Frames einteilen
- Reed-Solomon Checksumme an Frames anfügen
- 14-bit EFM-Daten aus 8-Bit Nutzdaten
- 17-bit EFM mit Merge-Bits
- 17-bit EFM, Sync-Pattern anfügen
- Pits and Lands

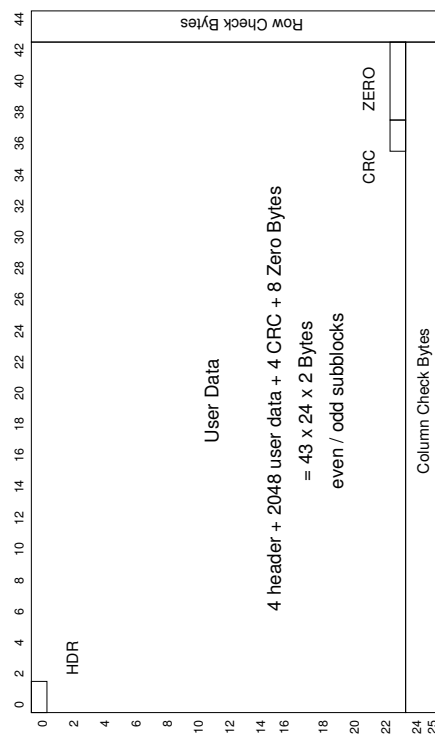


CD: Sektoren (Frames)



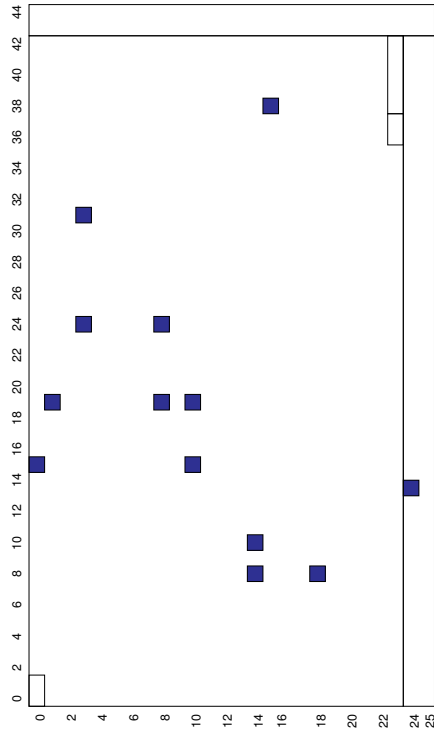
- 75 Sektoren pro Sekunde, 0 .. 74
- Numerierung per (minute, second, sector)
- 60 Minuten: 270.000 Frames (553 MB)
- 74 Minuten: 333.000 Frames (682 MB)

CD-ROM LEC Reed-Solomon Code



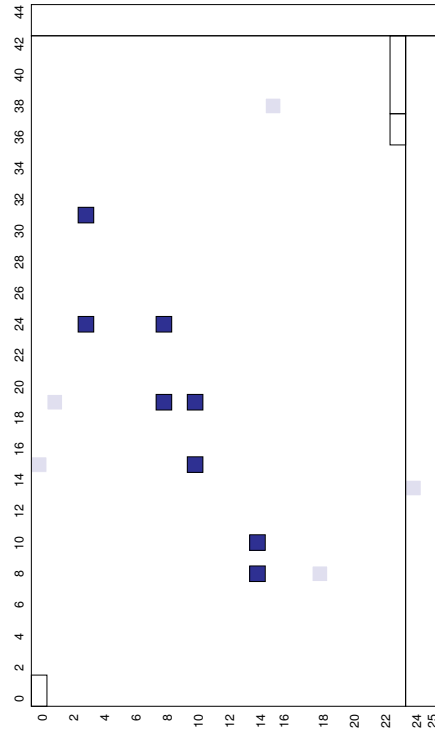
- Layered Error Correction (CD-ROM Mode 1)

CD: Reed-Solomon Code



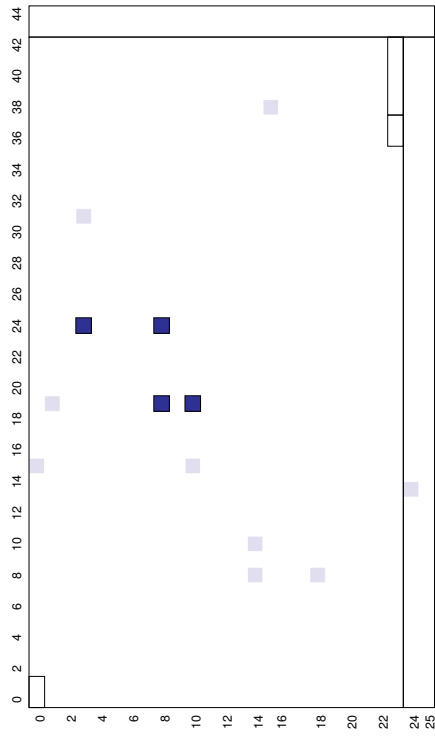
- Beispiel: Fehler vor der Korrektur

CD: Reed-Solomon Code



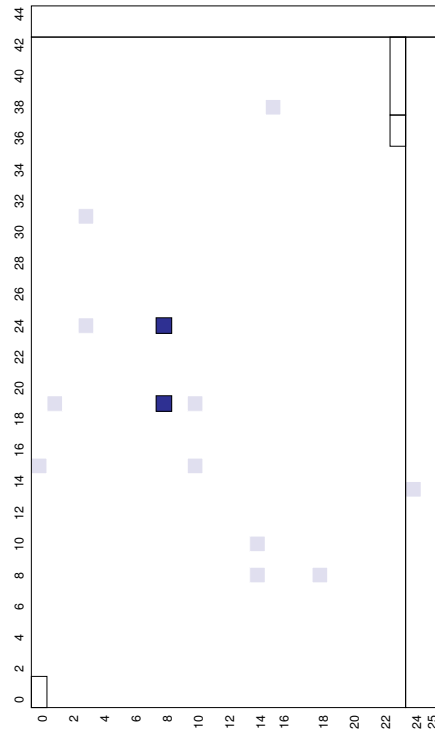
- erster Schritt: alle Einzelfehler in Zeilen korrigiert

CD: Reed-Solomon Code



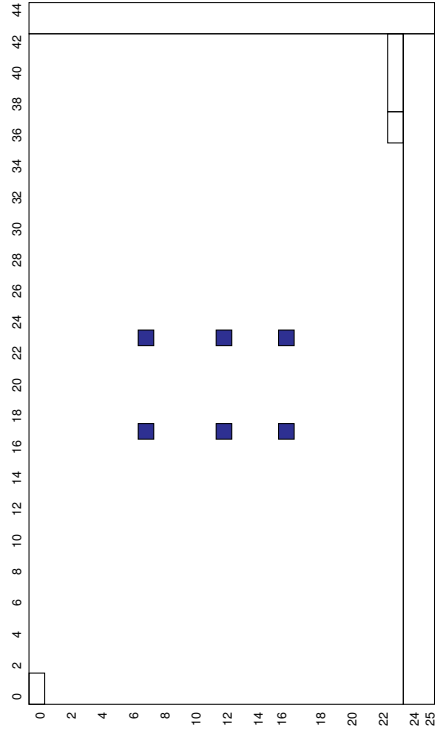
- zweiter Schritt: alle Einzelfehler in Spalten korrigiert

CD: Reed-Solomon Code



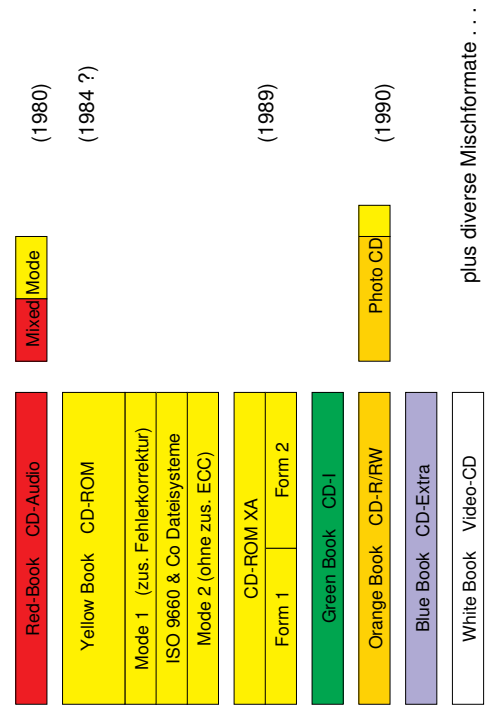
- dritter Schritt: wieder die Zeilen korrigiert, dann wieder die Spalten

CD: Reed-Solomon Code

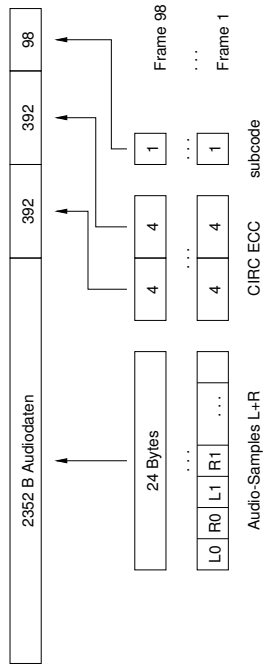


- keine Einzelfehler in Zeilen/Spalten, trotzdem korrigierbar
- Bitfehlerrate CD-ROM besser als 1E-13

CD: "colors"

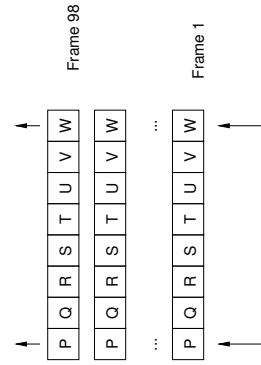


CD: Audioformat, konzeptionell



- 98 Frames a 24 Bytes pro Sektor
- 24 Bytes: je 6 Samples linker/rechter Kanal
- ein Byte Subcode pro Frame
- $75/s * 2352B = 44100/s * 16b * 2 / 8 = 176 \text{ KB/s}$
- ECC korrigiert Bursts bis zu 7000 fehlenden Bits

CD: Subchannels



Subchannels / Subcode:

- 1 Byte Subcode pro Frame
- P markiert Start eines Tracks
- Q für Inhaltsverzeichnis der CD (TOC)
- R ..W abhängig vom Format
z.B. konstant Null bei CD-ROM
Nutzung für CD-Text
- Datenrate: $75 * 98 \text{ bit/s} = 918 \text{ B/s}$

CD: Datenformate Daten / Audio

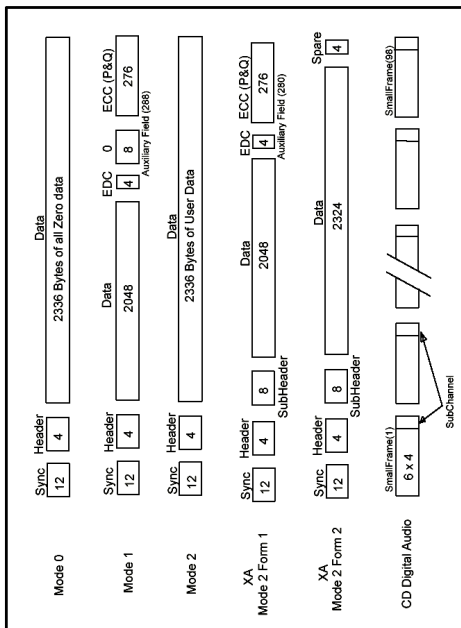
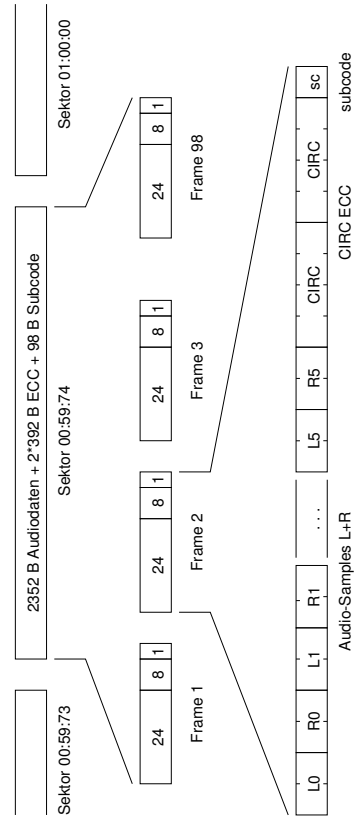


Figure 2 - CD-ROM Sector Formats

- Kapazität vs. Fehlerkorrektor

CD: Audioformat, Sektoren, Frames

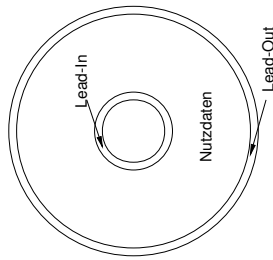


- 75 Sektoren pro Sekunde
- 98 Frames a 24 Bytes (+ECC) pro Sektor
- je 1 Byte Subcode pro Sektor

CD: Lead In / Lead Out

Lead-In:

- spezieller Bereich am Anfang (innen) der CD
- Nutzdaten konstant Null
- Q-Subcode enthält das Inhaltsverzeichnis
- bis zu 99 Tracks erlaubt: ca. 9 MByte
- Lead-Out kennzeichnet Ende der CD
- Nutzdaten und Subcodes konstant Null
- Multisession-CDs: je 1 Lead-In/Out Bereich pro Session plus Master-Lead In / Lead-Out



CD: Photo-CD

- Kodak / Philips 1993
- basiert auf CD-ROM/XA
- Filme (Kleinbild) werden mit 2000 dpi gescannt
- Auflösung 3072x2048 Pixel (optional 6144x4096)
- bis ca. 100 Photos
- multisession-Format (erlaubt mehrere Filme)
- proprietäres Datenformat
- mehrere Auflösungen: 192x128 bis 3072x2048 Pixel
- vergleichsweise hohe Kosten
- Markterfolg nur im Profi-Bereich
- neuer Versuch als "Picture-CD" (mit Intel/Adobe Software) 1024x1536 Pixel, JPEG-Format



CD: Mixed Mode CD

- kombiniert Audio-Tracks mit Daten/Video-Tracks
- Audio CD-Player erwarten nur Audio-Tracks:
 - ältere Player lesen Daten als Audio
 - Knacksen (Vorsicht: extreme Lautstärke)
 - neuere Player schalten den Track stumm
 - alternative Anordnung: Datentrack(s) ganz hinten
- "kranke" Block- bzw. Min/Sek/Frame-Adressierung (siehe Beispiel auf nächster Folie)
- wird von allen aktuellen PC-Laufwerken unterstützt

CD: Mixed Mode CD (Beispiel)

Block Description	Logical Address (Decimal)	Track Relative logical address	Absolute MS/F Address ¹	Track / Index	Track Relative MS/F Address	Sector Contains Info or Pause	Mode Audio or Data	CD Data Mode ²
Lead-in Area	---	---	---	0/-	---	---	Audio	---
Pre-gap	---	---	00/00/00	1/0	00/02/00 ⁷	Pause	Data	Null
1st Track data	0000 ⁴	0	00/02/00 ⁵	1/1	00/00/00	Info	Data	L-EC
2nd track data	6000 ⁴	0	01/22/00 ⁵	2/1	00/00/00	Info	Data	L-EC
Post gap	7500	1500	01/42/00	2/2	00/20/00	Info	Data	L-EC
Pause-silence	9000	3000	02/02/00	2/3	00/40/00	Pause	Data	Null
3rd track audio	9150	-150 ⁶	02/04/00	3/0	00/02/00 ⁷	Pause	Audio	---
	9300 ⁸	0	02/04/00 ⁸	3/1	00/00/00	Info	Audio	---
	1400	2250	02/34/00	3/2	00/03/00	Info	Audio	---
4th track audio	2197 ⁸	0	04/53/00 ⁶	4/1	00/00/00	Info	Audio	---
Pre-gap part 1	30000	-225 ⁶	06/40/00	5/0	00/03/00 ⁷	Pause	Audio	---
Pre-gap part 2	300075	-150	06/41/00	5/0	00/02/00 ⁷	Pause	Data	Null
5th track data	30225	0	06/43/00	5/1	00/00/00	Info	Data	L-EC
Last Information	263999 ¹⁰	233 774	58/39/74	5/1	51/56/74	Info	Data	L-EC
Post-gap	---	233 775	58/40/00	5/2	51/57/00	Pause	Data	Null
Lead-out area	264000 ¹¹	0	58/42/00	AA _n ¹³	00/00/00	Pause	Audio	---

CD: LBA/MSF Umrechnung

Table 207- LBA to MSF translation

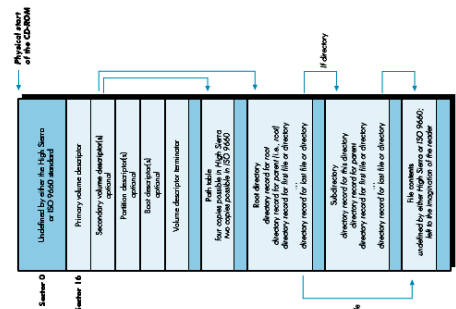
Condition	Formulae
$-150 \leq LBA \leq 404849$	$M = IP\left(\frac{LBA + 150}{60 \cdot 75}\right)$ $S = IP\left(\frac{LBA + 150 - M \cdot 60 \cdot 75}{75}\right)$ $F = IP(LBA + 150 - M \cdot 60 \cdot 75 - S \cdot 75)$
$-45150 \leq LBA \leq -151$	$M = IP\left(\frac{LBA + 450150}{60 \cdot 75}\right)$ $S = IP\left(\frac{LBA + 450150 - M \cdot 60 \cdot 75}{75}\right)$ $F = IP(LBA + 450150 - M \cdot 60 \cdot 75 - S \cdot 75)$
$00/00/00 \leq MSF \leq 89/59/74$	$LBA = (M \cdot 60 + S) \cdot 75 + F - 150$
$90/00/00 \leq MSF \leq 99/59/74$	$LBA = (M \cdot 60 + S) \cdot 75 + F - 450150$

- logische Blockadresse vs. Minute/Sekunde/Frame

CD-ROM: ISO 9660

Standard-Dateiformat für CD-ROMs

- Daten starten in Sektor 16 (00:02:16)
- DOS-kompatibel (FAT)
- Dateinamen mit 8+3 Zeichen
- bis zu 8 Verzeichnisebenen
- Level-2 erlaubt Namen bis 32 Zeichen
- plattformunabhängig
- Dateien müssen linear vorliegen
- keine späteren Änderungen möglich
- Level-3 erlaubt fragmentierte Dateien



CD-ROM: Joliet

- Microsoft-Erweiterung von ISO-9660
 - erlaubt Windows95-Dateinamen
 - Namen bis 64 Zeichen, inklusive Sonderzeichen
 - integriert in Windows 9x/2K
 - ebenfalls in neueren Linux-Versionen
- www.plateau.cs.berkeley.edu/people/chaffee/joliet.html

CD-ROM: Rock-Ridge und andere

Rock-Ridge:

- Erweiterung von ISO-9660 für Unix-Systeme
- erlaubt lange Dateinamen
- Unix-style Datei-Attribute (owner, permissions)
- symbolische Links
- abwärtskompatibel (ISO-9660 Systeme sehen die 8+3 Daten)
- ftp.yimi.com/pub/rockridge/

Macintosh HFS:

- CD-ROM Format mit Apple's hierarchical file system
- völlig inkompatibel mit ISO-9660
- benötigt entsprechende Treiber

CD-ROM: El Torito

bootfähige CD-ROMs?

- "El-Torito" Spezifikation von Phoenix und IBM (1994)
- Grundidee: BIOS ersetzt Laufwerk A: durch die CD-ROM
- basiert auf ISO-9660 Dateisystem
- Sektor 16 wie gehabt für Primary Volume Descriptor
- Sektor 17 als Boot Record Volume Descriptor
- erlaubt mehrere Boot-Sektoren pro CD
- Zugriff über BIOS/DOS INT-13 Schnittstelle
- CD-ROM kann als Live-Filesystem genutzt werden

CD-ROM: El Torito

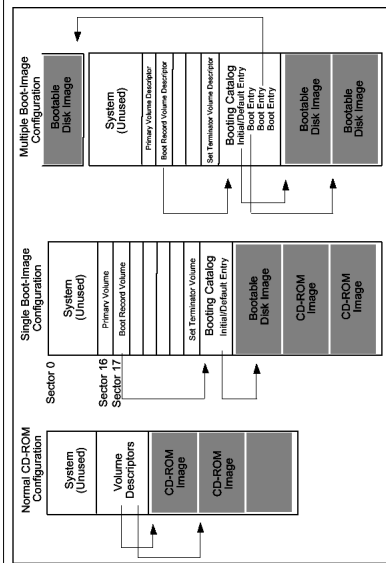
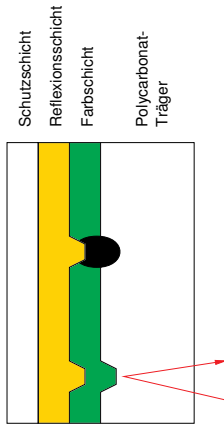


Figure 1. Three types of CD-ROM configuration:

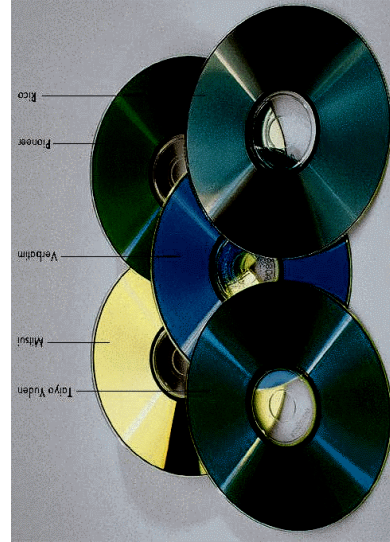
1. The Normal CD-ROM configuration is not bootable, uses Root Directory and CD-ROM drivers to access CD-ROM images.
2. A BIOS with Single Boot-Image capability accesses the Initial/Default Entry to access a single bootable disk image. After loading the operating system, the system can revert to standard CD-ROM drivers and the Root Directory to access CD-ROM images.
3. A BIOS with Multiple Boot-Image capability can access any one of a number of Bootable Disk Images listed in the Booting Catalog. After loading the operating system, the system can access other items in the disk image using INT 13 calls or return to normal access of CD-ROM images using CD-ROM drivers and the Root Directory.

CD-R: Prinzip



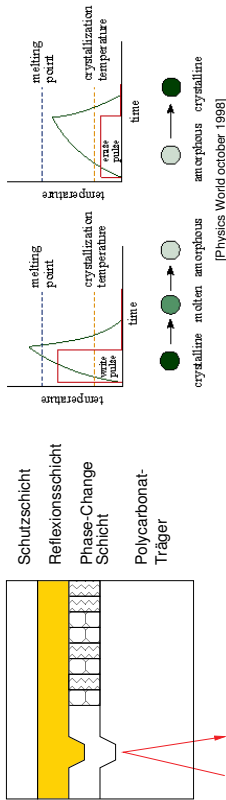
- mechanische Prägung (Pits/Lands) nicht praktikabel
- statt dessen: Farbstoff durch Laserimpuls zerstören
- etwas andere Reflexionsdaten als CD
- Spurführung des Pickups erfordert Daten: => Rohlinge enthalten vorbereitete Spiralspur (siehe DVD pre-groove)

CD-R: Rohlinge, Farbstoffe



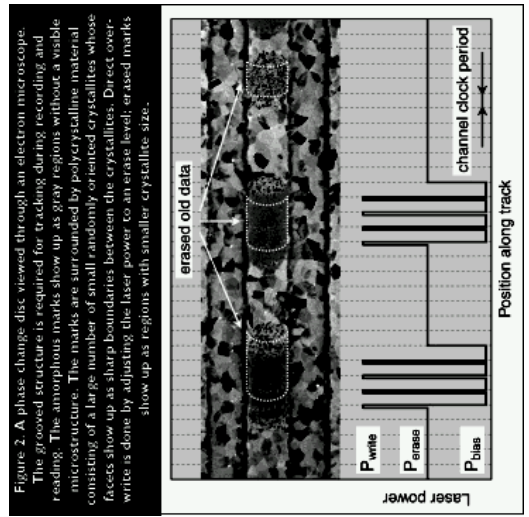
- diverse Farbstoffe, aber Haltbarkeit, Schreibigenschaften ähnlich

CD-RW: Prinzip



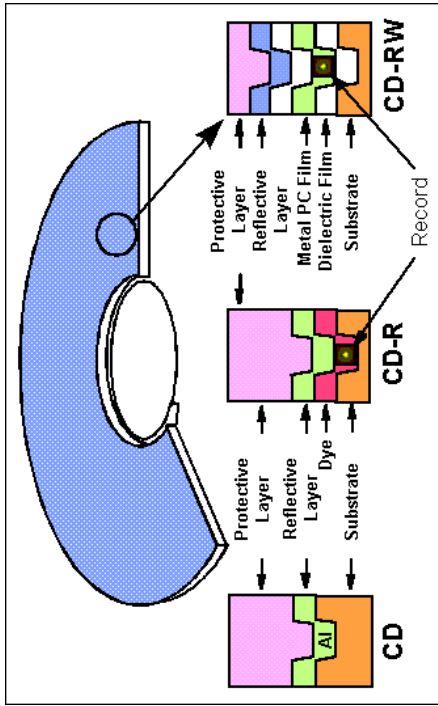
- Phase-Change Verfahren für wiederbeschreibbare CDs
- Material mit kristalliner / amorpher Struktur
- deutlich kleinere Reflexionsänderung als bei CD/CDR
- Umschalten durch schwache/starke Laserimpulse
- schnelle Abkühlung: amorph, langsame Abkühlung: kristallin
- bis zu 100.000 Mal wiederbeschreibbar (theoretisch)

CD-RW: amorph / polykristallin



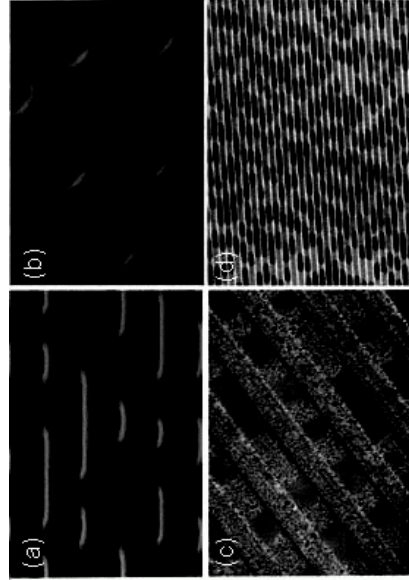
[ICAM 43-11]

CD-RW: Aufbau CD / CDR / CDRW



- CD-Pressung "parallel"
- CD-R / CD-RW Schreiben sequentiell, entlang der Rohnspur

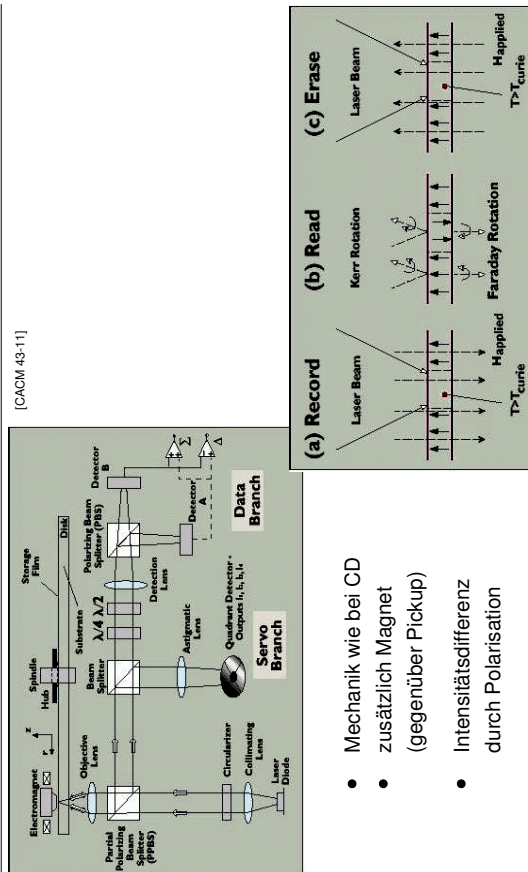
Pits: CD, CD-R, CD-RW, MO



- a) CD (Pits gepreßt)
- b) CD-R (Pits gebrannt)
- c) CD-RW (amorph/kristallin)
- d) MO (Kerr-Effekt)

[PhysicsWorld October 1998]

magneto-optische Verfahren (MO)



- Mechanik wie bei CD
- zusätzlich Magnet (gegenüber Pickup)
- Intensitätsdifferenz durch Polarisierung

CD-R: erweitertes Lead-In

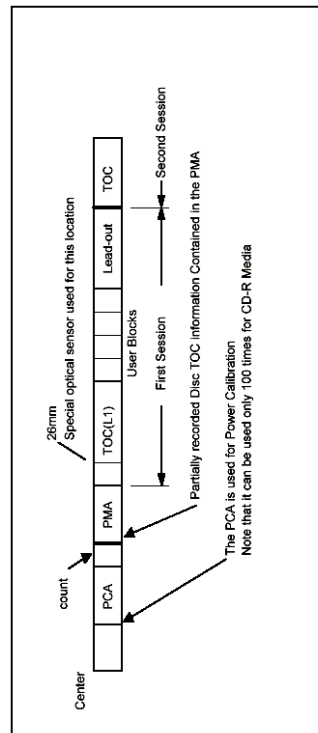


Figure 3 - CD-R/RW Disc Layout

- erweiterte Lead-In Zone (weiter innen als normale CD)
- u.a. Kalibrierung der Laserintensität beim Schreiben
- Audio/Datenformat unverändert

CD-R: "Überbrennen"

- Länge der Rohspur definiert die Kapazität der CD-R/RW
 - spezielle Rohlinge (80 Min) mit engerer Rohspur
- "Überbrennen":
- angegebene Kapazität enthält >90 Sekunden Lead-Out
 - plus einige Sekunden Reserve
 - verkürztes Lead-Out erlaubt mehr Daten
 - muß von Brenner und Software unterstützt werden (zB. www.feurio.com)
 - evtl. Probleme mit älteren / abgenutzten Playern
 - alternativ für Audio: Daten minimal stauchen

CD-R: Buffer-Underrun

- sequentielles Schreiben der CD-R:
- => Brenner benötigt kontinuierlichen Datenstrom typische Puffergröße 2..4 MB
- Problem Buffer-Underrun:
- CD-R entspricht nicht mehr den Normen
 - Rohling defekt (CD-R) / neu formatieren (CD-RW)
- www.burn-proof.com:
- Brenner rechtzeitig (kontrolliert) stoppen
 - Position auf der CD-R merken (Spur, Position 100 µm)
 - neu aufsetzen, sobald Daten verfügbar
 - Fehlerkorrektur beseitigt die Lücke ("burst error")
 - wird von einigen neuen Brennern unterstützt

CD-R: Audio-Grabbing

digitales Auslesen von CD-DA:

- optimal mit Audio-Playern (Digitalausgang, 1X Speed)
 - Digitalausgänge an CD-ROMs selten / oft fehlerhaft
- "Packet"-Interface problematisch:
- in alten Laufwerken schlecht implementiert
 - Audio-Format hat keine fortlaufenden Sektor-IDs
 - mm:ss:ff-Marken: ff-Werte fehlen manchmal
 - Packet vs. Streaming: Probleme beim Wiederaufsetzen
 - nur einfache Fehlerkorrektur, kein LEC
- => gutes Laufwerk notwendig
=> mehrfaches Lesen / Korrelation der Daten (cdparanoia)

CD: Audio Grabbing via SCSI3 MMC

Table 95 - CD-DA (Digital Audio) Data Block Format

Bit	7	6	5	4	3	2	1	0
Byte								
0	Left Channel (Lower Byte) (LSB)							
1	(MSB)							
2	Right Channel (Lower Byte) (LSB)							
3	(MSB)							
2348	Left Channel (Lower Byte) (LSB)							
2349	(MSB)							
2350	Right Channel (Lower Byte) (LSB)							
2351	(MSB)							

If the CD Drive does not support the CD-DA Stream-Is-Accurate capability, See Table 230 - CD Capabilities and Mechanical Status Page, then the digital audio data must be read as a continuous stream. If while streaming the drive must stop, there will be a non recoverable error generated READ ERROR - LOSS OF STREAMING. This is due to the 1 second uncertainty of the address. (i.e. there is no header in CD-DA data). Reissuing the command may not return exactly the same data as the previous try. When the drive supports the stream accurate capability, there will be no error, only some time delay for rotational latency.

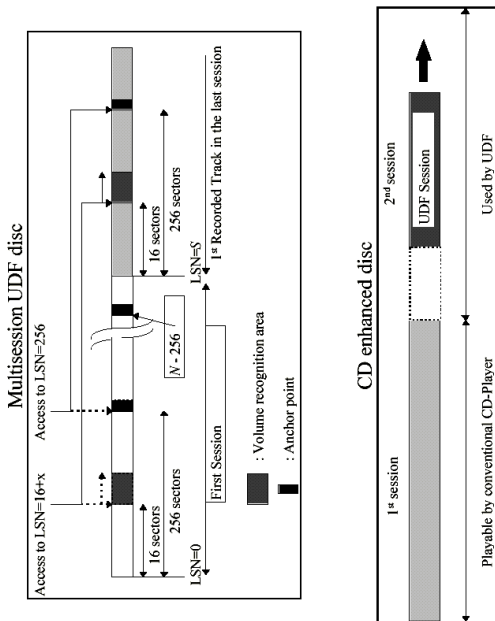
UDF: Dateisystem

- CDR Medien sind nur einmal beschreibbar
 - ISO-9660 erwartet TOC und Directories an fester Position
- => spätere Änderungen unmöglich
- UDF-Dateisystem: "universal disk filesystem"
- basiert auf ISO 9660
 - aber erweitertes, flexibleres Dateisystem
 - "virtual allocation tables"
 - gültiges Directory jeweils im letzten geschriebenen Block
 - dort Verweise auf Dateien und ältere Directory-Blöcke
 - keine Beschränkung der Verzeichnis-Schachteltiefe
 - Finalisieren der CD erzeugt volles ISO 9660 Dateisystem
 - www.osta.org

UDF: Packet Writing

- CDR Medien sind nur einmal beschreibbar
- UDF-Packet Writing:
- Dateien in einzelnen kleinen Paketen schreiben
 - zunächst ohne TOC im Lead-In
- "virtual allocation tables":
- gültiges Directory jeweils im letzten geschriebenen Block
 - dort Verweise auf Dateien und ältere Directory-Blöcke
 - Dateien können immer noch nicht gelöscht werden
 - neues Directory ohne Verweis auf gelöschte Datei schreiben
 - Datei modifiziert:
 - neue Datei schreiben, neues Directory schreiben

UDF: Multisession / enhanced disks



DVD: Konzept und Anforderungen

Anforderungen für DVD-Video:

- 135 Minuten Spieldauer pro Seite
- bessere Auflösung als die Laserdisc
- Surround-Audio in CD-Qualität
- Audiospuren für 3-5 Sprachen
- Untertitel in mehreren Sprachen
- diverse Bildformate (Letterbox, Pan, Widescreen)
- Interaktion wie bei Video-CDs
- Jugendschutz
- Kopierschutz
- CD-kompatibel
- Herstellungskosten ähnlich wie CDs

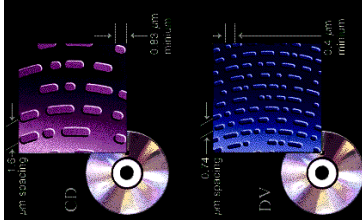
MPEG-2, AC3:
=> ca. 6 Mb/s
=> 4-5 GB / Seite

DVD: Verbesserungen

höhere Kapazität der DVD gegenüber der CD:

- kleinere Pits, kleinerer Spurbestand
- veränderte Header-Strukturen
- weniger Parity-Bits
- Weglassen der Subcodes
- 2048-Byte Sektoren
- kleinerer nicht-genutzten Innenteil (Radius CD 25 mm, DVD 24 mm)

=>	DVD-5	single layer	4.4 GB	6.7x
	DVD-9	double layer	8.0 GB	12.3x
	DVD-10	double side	8.8 GB	13.5x
	DVD-18	DS / DL	15.9 GB	24.4x CD-ROM



DVD: Vergleich mit CD und DVR

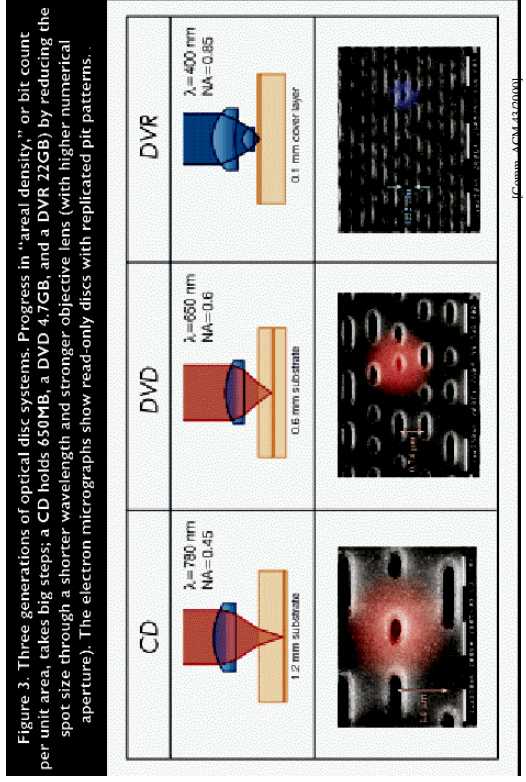
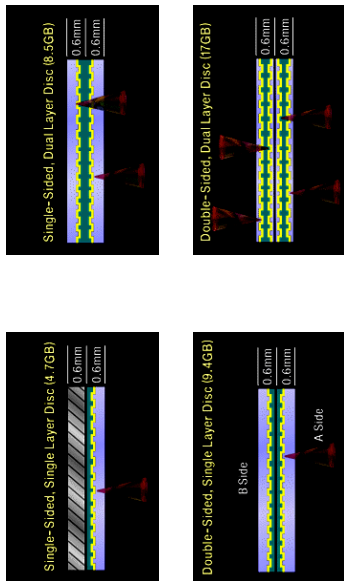


Figure 3. Three generations of optical disc systems. Progress in "areal density," or bit count per unit area, takes big steps; a CD holds 650MB, a DVD 4.7GB, and a DVR 22GB) by reducing the spot size through a shorter wavelength and stronger objective lens (with higher numerical aperture). The electron micrographs show read-only discs with replicated pit patterns.

DVD: 4 Formate



- single/double side
- single/dual layer (äußere Schicht halbdurchlässig)

DVD: Datenformat (Header)

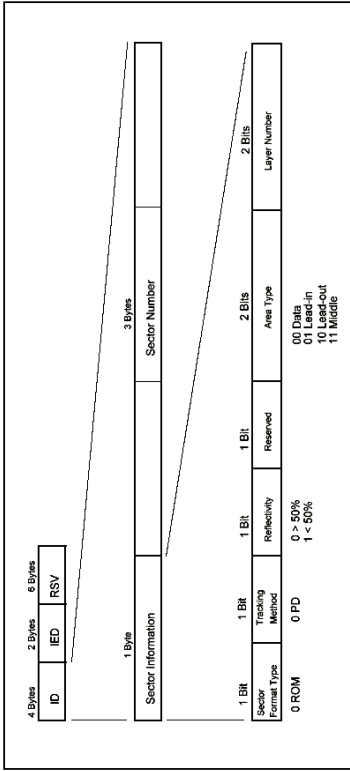


Figure 11 - Header Layout

- vollständige Information für jeden einzelnen Block
- 2-bit layer number: Seite 1/2, außen/innen
- vergleiche CDROM

DVD: Datenformat (Sektorformat)

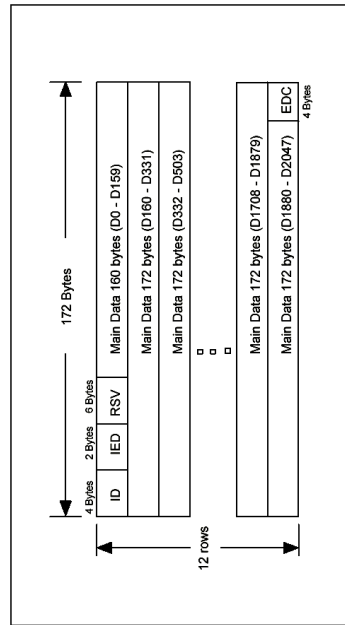


Figure 13 - Sector Layout

- eindeutige Block-ID, 4-Byte layered ECC
- vergleiche CDROM

DVD: Sektoranordnung dual-layer

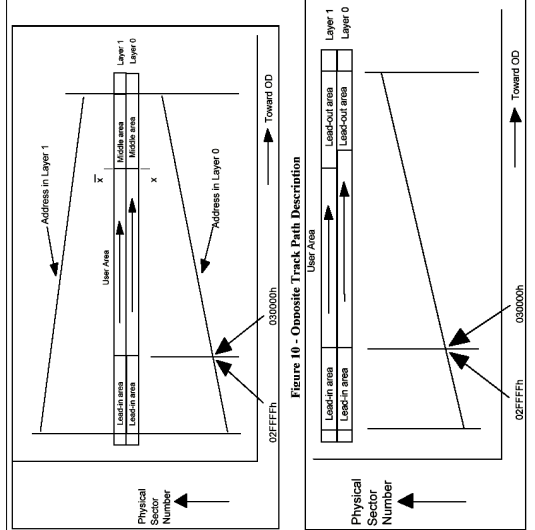
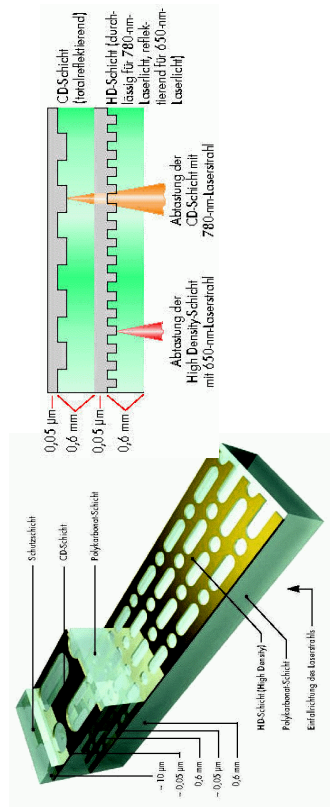


Figure 9 - Parallel Track Path Description

Figure 10 - Opposite Track Path Description

DVD: Super Audio CD



- Kompatibilität mit Audio-CDs
- zweite (DVD)-Schicht mit Stereo "bitstream", 2,8 Mb/s
- vgl. DVD-Audio

[ct 21/98 242]

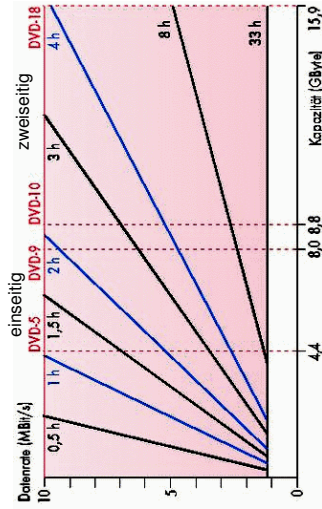
DVD-Audio

- Spezifikation für DVD-Audio seit Q1/1999
- nutzt die DVD-5 (4.7 GB)

diverse Audioformate werden unterstützt:

- Abstraten 44.1 / 48 / 88.2 / 96 KHz
- Quantisierung mit 12 / 20 / 24 bit
- mindestens 74 min. Spieldauer für alle Modi
- 16 bit, 44.1 Stereo, 7 Stunden Spieldauer
- 24 bit, 96 KHz, 2-6 Kanäle Surround
- 24 bit, 192 KHz Stereo
- Dolby Digital, DTS, MPEG-AAC, ...
- plus Standbilder und Textinformationen
- bisher kaum erhältlich

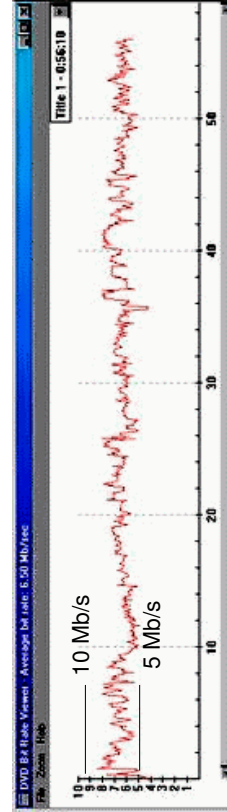
DVD: Video, Datenrate vs. Spieldauer



- Formate: 720x576x25 PAL / 720x480x29.97 NTSC
- 2 Stunden Spieldauer gewünscht, bei 5 GB
- typische Datenrate für MPEG-2 mit AC3-Audio
- Digitales Fernsehen: DF1 sendet MPEG-2 mit

- 5.5 Mb/s
- 1.5 .. 9.8 Mb/s
- 6.8 Mb/s

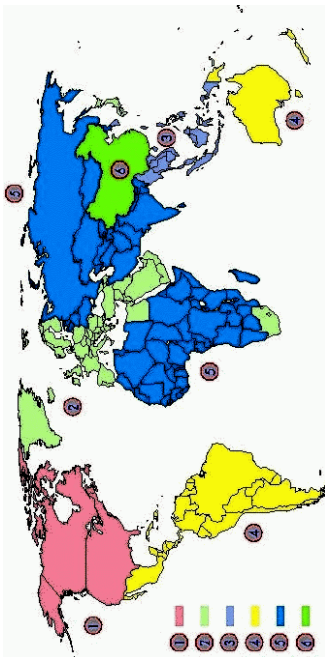
DVD: Datenrate MPEG-2



[ct 20/99, Sesamestreet, Region 0]

- typische Datenrate der Video-DVD ist 5 .. 10 Mb/s
- andere Datenformate (MPEG-4) bisher nicht verwendet
- siehe MPEG Standards

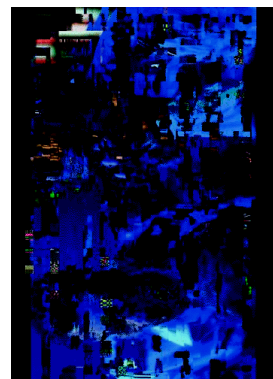
DVD: Region Codes



DVD-Video spezifiziert Region-Codes

- zeitversetzte Veröffentlichungen zwischen USA / EU / Japan
- Sicherung des Kino-Marktes
- Region 0 ist universell nutzbar
- Code in Laufwerks-Firmware, typisch höchstens 5x wechselbar

DVD: CSS



- direkte Kopie einer DVD-Video
- Daten größtenteils unlesbar
- nicht alle DVDs sind verschlüsselt

"Content Scrambling System":

- Schutz vor digitalen (=perfekten) Raubkopien
- verschlüsselte Übertragung zwischen Laufwerk und Decoder (HW/SW)
- komplexes Challenge-Response-Protokoll zur Authentifizierung
- Codes im Lead-In der DVD gespeichert, dort nicht zugreifbar
- Verfahren nicht publiziert, nur für Hersteller zugänglich

DVD: DeCSS

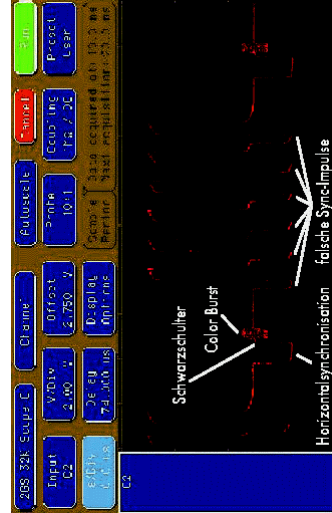
mittlerweile ist CSS gecrackt:

- diverse Angriffspunkte in den Windows Treiberschichten
- Screenshots -> AVI
- Software-Player cracken
- algorithmische Angriffe möglich wegen 40-bit Schlüssel

DeCSS:

- Windows-GUI
 - dekodiert DVD-Daten auf die Festplatte
 - verwendet Player-Key aus Xing Software-Player
 - Verbreitung via Internet / Abmahnungen durch DVD-Anwälte
- => ermöglicht DVD-Player für Linux
=> rechtliche Situation unklar

DVD: Macrovision



Schutz gegen analoge DVD-Kopien auf Videorekorder:

- zusätzliche Synchronimpulse
- AGC: wechselnde Schwarzschulter-Werte
- im unsichtbaren Bereich: Fernseher ignoriert das Signal

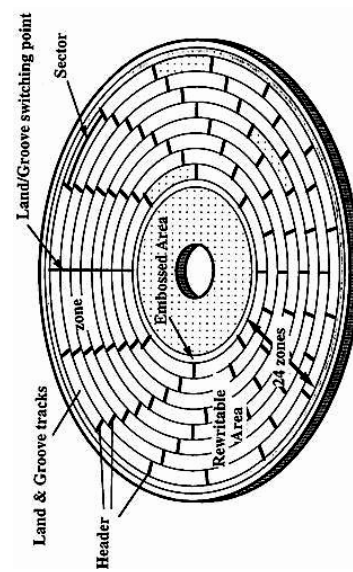
DVD: DVD-RAM

wiederbeschreibbare DVDs:

- mehrere, untereinander inkompatible Verfahren
- nicht mit DVD-R kompatibel
- Phase-Change-Technik wie CD-RW
- zoned CLV, wobbled pre-groove, usw.

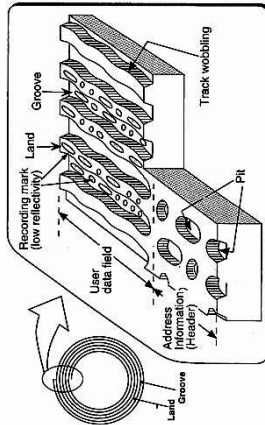
	Ver 1.0	Ver 1.9 / 2.0
Seiten	1 / 2	1 / 2
Kapazität verfügbar	2.6 GB	4.7 GB
Pit µm	0.41 x 0.74	0.28 x 0.615
Verfahren	phase change wie CD-RW	

DVD: DVD-RAM Sektoren



- Disk ist in 24 Zonen eingeteilt
- innerhalb einer Zone konstante Umdrehungsgeschwindigkeit

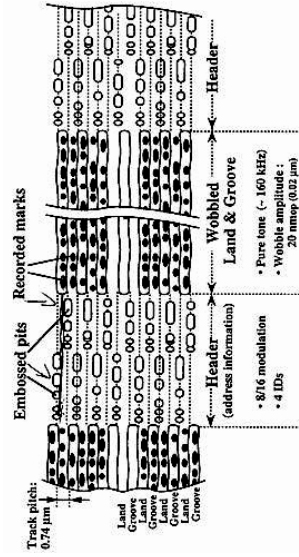
DVD: DVD-RAM Pregroove



CD/DVD-Rohlinge enthalten eine Roh-Spur:

- Roh-Spur für Spurführung des Schreib/Lese-Kopfes
- Track-Wobbling für Drehzahlregelung
- regelmässig Header-Sektoren
- DVD-RAM: Daten abwechselnd in Lands und Grooves

DVD: DVD-RAM Datenaufzeichnung



- Rohlinge enthalten fertige Header-Zonen
- DVD-RAM Typ, Kapazität, Schreibgeschwindigkeit, usw.
- Aufzeichnung nur in die Datenbereiche
- UDF: max 1 Partition a 2.3 GB / Seite
- FAT 16: mehrere Partitionen a 2 GB / Seite

Datenformate

effiziente Speicherung und Übertragung von Audiodaten?

- unkomprimierte Darstellung, PCM
- WAV-Format
- ADPCM
- Sprachcoders, Kompanidierung
- S/PDIF und ADAT-Schnittstellen

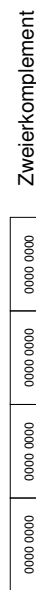
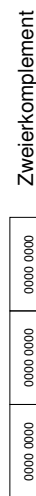
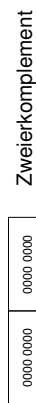
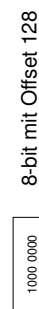
spätere Themen:

- psychoakustische Verfahren
- Streaming
- MIDI

Datenformate: Zahldarstellung

geeignete Zahldarstellung für Audiodaten ?!

- 8/16/24-bit Integer, Offset oder Zweierkomplement
- 32-bit Gleitkomma
- Bitströme



31 24 23 16 15 8 7 0
Konvention: -1 <= x <= +1

Datenformate: Kodierung

"direct coding"	PCM
"entropy coding"	repetitive sequence suppression
	statistical encoding
"source coding" (e.g. speech)	transform encoding
	differential encoding
	vector quantization
	zero suppression
	run-length encoding
	pattern substitution
	Huffman encoding
	FFT
	DCT
	...
	DPCM
	delta modulation
	ADPCM
	general / fractal / ...

Datenformate: Entropie/Quellenkodierung

Entropiekodierung:

- Eigenschaften der Datenquelle werden ignoriert
- Signalwiederholungen entfernen
- statistische Verfahren, z.B. Huffman-Kodierung
- verlustfrei und reversibel
- für Audiodaten: ca. Kompressionsfaktor 2 erreichbar

Quellenkodierung (source encoding):

- Eigenheiten der Datenquelle / senke berücksichtigen
- z.B. Frequenzgang / Maskierung / Rauschschwellen des Ohrs
- verlustfrei
- verlustbehaftet für bessere Kompression, z.B. MP3 bis ca 10:1

Datenformate: SND

```
typedef struct {
int magic;
/* 0x2e736e64 = ".snd" */
/* offset to the data */
int dataLocation;
/* number of bytes of data */
int dataSize;
/* l=1-law, 2= linear8, ... */
int dataFormat;
/* samples per second */
int samplingRate;
/* channelCount;
char info[4];
/* optional text info */
} SNDSoundStruct;
```

- erstes Audioformat auf NeXT und Sun
- einfache Dateistruktur mit Kopf (SNDSoundStruct) und Daten
- übliche Datenrate: 8-bit mono, 8 KHz Samplerate
- diverse Datenformate von 8-bit linear bis G.723
- Zugriff über entsprechende API (NeXT Sound Kit)

Datenformate: WAV

- Standard-Dateiformat für Audiodaten unter Windows
- Abkömmling des EA IFF85 Formats

"Chunk"-Format:

- Datei besteht aus einzelnen "Häppchen"
- jeder Chunk enthält eigene Headerinformation
- und optional Daten
- Format kann nachträglich um neue Chunks erweitert werden

Zugriff auf hintere Chunks:

- durch Verkettung der Länge der vorherigen Chunks
- erfordert Kenntnis aller vorangegangenen Chunks
- ungeeignet für Streaming / verlustbehaftete Kanäle

Codec

- verlustlose Datenkompression:
 - $decode(encode(x)) = x$
 - jedes Verfahren benötigt Paar aus Coder und Decoder
 - := "CODEC"
- vollständige Abstraktion und Kapselung möglich:
- $decode1(decode2(kanal(encode2(encode1(x))))) = x$
 - äußere / innere Schichten brauchen nicht vom Codec zu wissen
 - beliebig tiefe Schachtelung
 - siehe Windows / Java Media Framework

Codecs: unter Windows 9x



Windows-Systemsteuerung:

- sammt CODECS
- Audio, Video, MIDI
- je nach installierter SW
- hier: 14 Audio-Codecs
- Sprache vs. Musik

Datenformate: WAV

Hierarchie mit Unzahl von Chunk-Typen:

- RIFF Chunk (Wave File Header)
- Format Chunk (Struktur des Data Chunks)
- Data Chunk (Daten, z.B. PCM Samples)
- Fact Chunk (Info über komprimierte Daten)
- Cue Chunk (Offset zu wichtigen Zeitpunkten)
- Playlist Chunk (Anspielfolge von Cuepunkten)
- Associated Data Chunk (z.B. Songtitel)
- Label Chunk (eigentlicher Titel)
- ...

- oft nur drei Chunks: Header/Format/Data
- alle Chunks "word-aligned", evtl. ein Füllbyte 0 ergänzen

Datenformate: WAV header

```
typedef struct {
    ID ckID; /* 0x52494646 = "RIFF" */
    long ckSize; /* file size - 8 */
    ID fmtType; /* 0x57415645 = "WAVE" */
    char pad[1]; /* padding, if ckSize odd */
} WaveChunk;

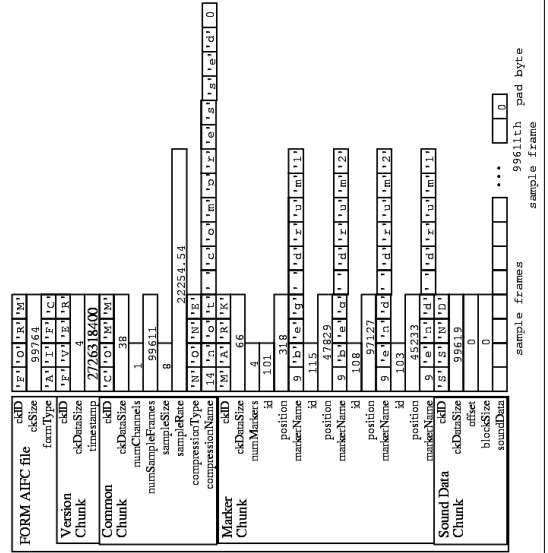
typedef struct {
    ID ckID; /* 0x666D7420 = ".fmt" */
    long ckSize; /* 16 + extra format bytes */
    short wFormatTag; /* e.g. WAVE_FORMAT_PCM */
    ushort nChannels;
    ushort nSamplesPerSec;
    ushort nAvgBytesPerSec;
    ushort nBlockAlign;
    ushort nBitsPerSample;
} FormatChunk;
```

Datenformate: WAV data

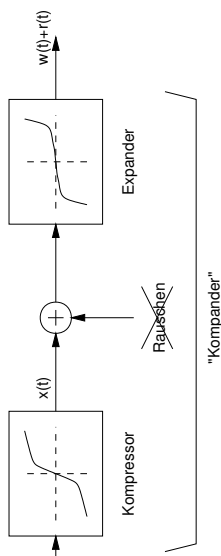
```
typedef struct {
    ID ckID; /* 0x64617461 = "data" */
    long ckSize; /* in bytes */
    char[] data;
}
```

- WAVE_FORMAT_PCM: 16-bit Zweierkomplementdaten
- aber auch G.711 / G. 721 / GSM / MPEG implementiert
- Stereo/Mehrkanaldaten als Frames
- links / rechts, bzw. Kanal 1, 2, 3, ...
- erlaubt das Abspielen, ohne die Datei komplett laden zu müssen

AIFF: Beispiel



G.711: Kompander



- digitale Übertragung (ISDN) ist rauschfrei
trotzdem Einsatz der Kompandierung:
- Erhöhung des Dynamikbereichs für die Sprachübertragung
- bzw. Reduzierung des Quantisierungsrauschens (leise Signale)
durch Anwendung der Kompressor-Kennlinie

μ -Law, α -Law

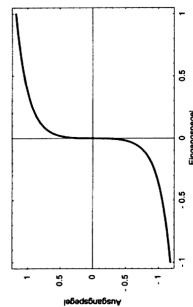


Bild 5.3: μ -Law-Kompressionskennlinie für normierte Signalpegel

μ -Gesetz (μ -Law)
Diese Art der Dynamikkompression ist im angelsächsischen Raum verbreitet, sie wird manchmal irreführend als μ -Law bezeichnet. Der zu digitalisierende Wert ergibt sich aus dem Eingangsspegel S (auf 1 normierter Signalpegel) als

$$S' = \frac{\text{sig}(\mu S) \cdot \ln(1 + \mu \cdot \text{abs}(S))}{\ln(1 + \mu)} \quad \mu = 255$$

- Kompression von 12..16 bit linear auf 8 bit
- Idee: logarithmische statt linearer Kodierung
- Berechnung: stückweise lineare Approximation / Tabellen

α -Law: lineare Approximation

```

/*
 * linear2alaw() - Convert a 16-bit linear PCM value to 8-bit A-law
 *
 * Linear Input Code          Compressed Code
 * -----
 * 0000000wxyz              000wxyz
 * 0000001wxyz              001wxyz
 * 0000010wxyz              010wxyz
 * 0000011wxyz              011wxyz
 * 0001000wxyz              100wxyz
 * 0001001wxyz              101wxyz
 * 0100000wxyz              110wxyz
 * 0100001wxyz              111wxyz
 *
 * For further information see John C. Bellamy's Digital Telephony, 1982,
 * John Wiley & Sons, pps 98-111 and 472-476.
 */

```

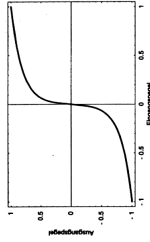


Bild 5.2: α -Law-Kompressionskennlinie für normierte Signalpegel

α -Law: Umrechnung

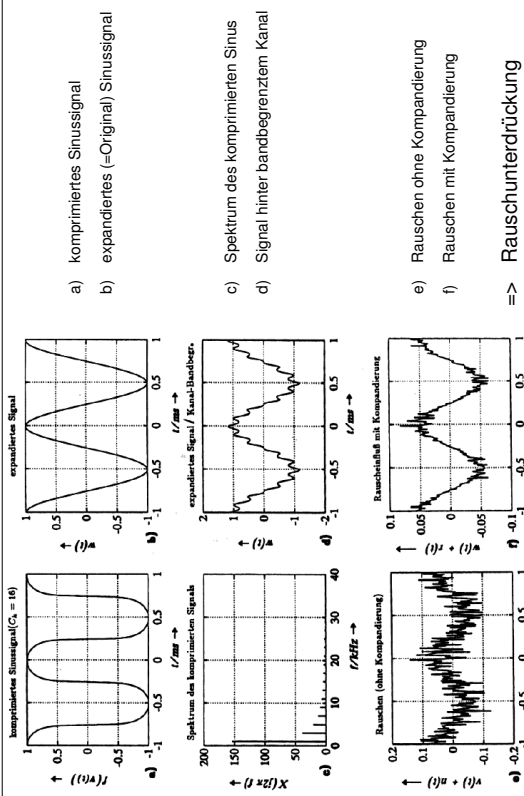
```

unsigned char_a2u[128] = {
1, 3, 5, 7, 9, 11, 13, 15,
16, 17, 18, 19, 20, 21, 22, 23,
24, 25, 26, 27, 28, 29, 30, 31,
32, 32, 33, 33, 34, 34, 35, 35,
36, 37, 38, 39, 40, 41, 42, 43,
44, 45, 46, 47, 48, 48, 49, 49,
50, 51, 52, 53, 54, 55, 56, 57,
58, 59, 60, 61, 62, 63, 64, 64,
65, 66, 67, 68, 69, 70, 71, 72,
73, 74, 75, 76, 77, 78, 79, 79,
80, 81, 82, 83, 84, 85, 86, 87,
88, 89, 90, 91, 92, 93, 94, 95,
96, 97, 98, 99, 100, 101, 102, 103,
104, 105, 106, 107, 108, 109, 110, 111,
112, 113, 114, 115, 116, 117, 118, 119,
120, 121, 122, 123, 124, 125, 126, 127};

```

- Konvertierung zwischen α -Law und μ -Law mit Tabellen
- beträchtliche Rundungs/Quantisierungsfehler

μ-Law: Kompendierung



GSM

"Global Standard for Mobile Communication"

- spezieller Codec für Sprachkodierung
- gute Sprachverständlichkeit
- aber völlig ungeeignet für Musiksignale
- möglichst einfacher Dekoder (Mobilgeräte)
- mehrere Datenraten: full / enhanced-full / half-rate
- z.B. full-rate mit 13.3 kb/s
- Dokumentation und Demo-Code: <http://kbs.cs.tu-berlin.de/~jutta/toast.html>

Sprach-Codex: Sprachqualität?

Gauging the speech quality is an important but also very difficult task. The signal-to-noise ratio (SNR) is one of the most common objective measures for evaluating the performance of a compression algorithm. This is given by:

$$SNR = 10 \log_{10} \left\{ \frac{\sum_{n=0}^M s^2(n)}{\sum_{n=0}^M (s(n) - \hat{s}(n))^2} \right\} \quad (1)$$

where $s(n)$ is the original speech data while $\hat{s}(n)$ is the coded speech data. The SNR is a long term measure for the accuracy of speech reconstruction and as such it tends to "hide" temporal reconstruction noise particularly for low level signals. Temporal variations of the performance can be better detected and evaluated using a short-time signal-to-noise ratio, i.e., by computing the SNR for each N-point segment of speech. A performance measure that exposes weak signal performance, is the segmental SNR (SEGSNR) which is given by

$$SEGSNR = \frac{10^{L-1}}{L} \sum_{l=0}^{L-1} \log_{10} \left\{ \frac{\sum_{n=0}^{N-1} s^2(iN+n)}{\sum_{n=0}^{N-1} (s(iN+n) - \hat{s}(iN+n))^2} \right\} \quad (2)$$

Sprach-Codex: Bitrate vs. MIPS

Algorithm	Bit Rate (bits/sec)	MOS/DRT/DAM	MIPS*	References
PCM (G.711)	64k	4.3/95/73	0.01	[150][152]
ADPCM (G.721)	32k	4.1/94/68	~2	[22][92][150]
LD-CELP (G.728)	16k	4.0*/-/	~19	[95][98]
RPE-LTP (GSM)	13k	3.47*/-/	6	[119][307]
Skyphone-MPLP	9.6k	3.4*/-/	11	[25]
VSELP (IS-54)	8k	3.45*/-/	13.5	[70][100]
CELP (FS1016)	4.8k	3.2/93.7/62.2	16	[30][78]
STC-1	4.8k	3.52/92.7/63.	13	[210][212][213]
IMBE	4.15k	3.4*/-/	3	[26][112][141]
STC-2	2.4k	2.9/90.1/56	13	[210][212][213]
LPC-10e (FS 1015)	2.4k	2.3/89.9/52.3	~7	[77][301]
LPC-LSP	800	~191.2*/-	~20	[166]

* estimated, + low score reported, processor dependent

The above complexity and performance figures were obtained from different sources and correspond to different implementation platforms and test environments. Therefore the performance and complexity figures do not always constitute an absolute measure for comparison.

Verlustfreie Datenkompression

Grundidee:

- lineare Prädiktion
- Vorhersage des Zeitverlaufs aus früheren Samplewerten
- Ziel ist die Minimierung der Leistung des Differenzsignals
- anschließende Entropiekodierung
- häufige Abtastwerte mit kurzen Datenworten kodieren
- Rahmenbildung
- Kompressionsraten bis ca. Faktor 2 erreichbar
- Rauschen / rauschartige Klänge verhindern höhere Werte
- Mathematik und Beispiel: siehe Zölzer, Kap. 9.1

Digitale Audioverarbeitung | WS 2000 | 18.205

Audio-Packprogramme: ZAP

- Zero Loss Audio Packer
www.emagic.de
- Packprogramm, optimiert für Audiodaten
- Entpacker frei erhältlich
- Kompressionsraten typisch 20% bis 60%
- aber Algorithmen nicht publiziert
- diverse Freeware-Programme verfügbar
- z.B. L TAC (linear transform audio coding)
www.ft.ee.tu-berlin.de/~liebchen/ltac.html

Digitale Audioverarbeitung | WS 2000 | 18.205

ADPCM: Beispiel

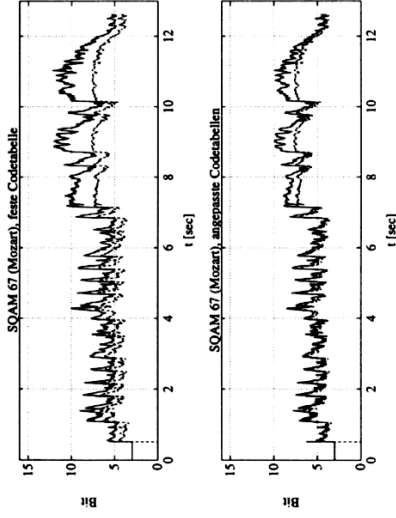


Bild 9.2: Verlustlose Datenkompression (Mozart): Wortbreite [Bit] über der Zeit (Entropie - -, lineare Prädiktion mit Huffman-Codierung —)

[Zölzer]

Digitale Audioverarbeitung | WS 2000 | 18.205

ADPCM: Beispiel

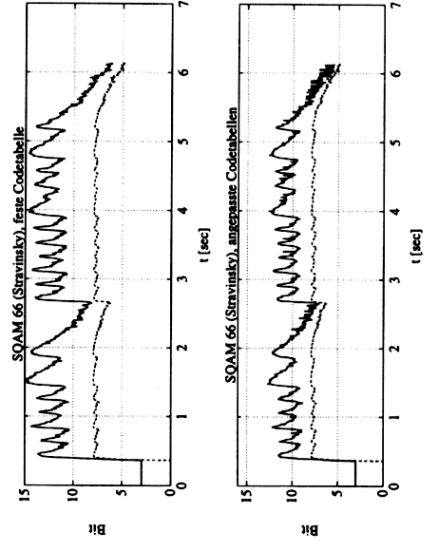


Bild 9.3: Verlustlose Datenkompression (Stravinsky): Wortbreite [Bit] über der Zeit (Entropie - -, lineare Prädiktion mit Huffman-Codierung —)

[Zölzer]

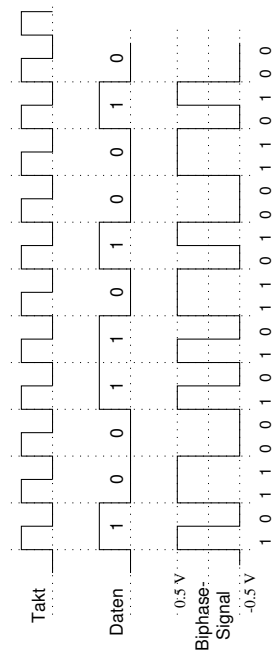
Digitale Audioverarbeitung | WS 2000 | 18.205

digitale Datenübertragung: SPDIF

Sony/Philips Digital Interface:

- standardisiert als IEC958 (1989-03 consumer part) AES/EBU (Audio engineering society, european broadcast union)
- diverse Status/Copyright/Zusatzinformationen
- "Consumer"-Variante mit SCMS-Kopierschutz
- unidirektionale, digitale Schnittstelle für Audiodaten
- optisch (Glasfaser) oder elektrisch (75-Ohm Koaxkabel)
- ein Stereo-Kanal, 16/20/24-bit Auflösung
- Samplerate 48 KHz, 44.1 KHz, 32 KHz
- festes Datenformat mit 192 Samples / Frame
- Bitrate (48KHz): $2 \cdot 32 \cdot \text{bit} \cdot 48000/\text{s} = 3.072 \text{ Mb/s}$
- aktuell: neue Mehrkanal-Datenformate (Dolby AC3, DTS)

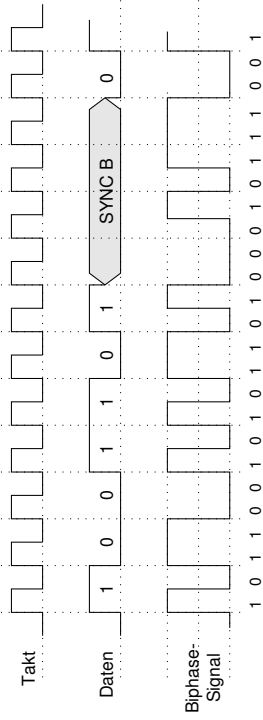
SPDIF: Kodierung



"Biphase"-Kodierung:

- ein Datenbit pro Takt, unterteilt in zwei "cells"
- Logikpegel wechselt in der ersten Zelle jedes Bits (Ausnahme: Sync)
- für jeden '1' Wert weiterer Wechsel in der zweiten Zelle
- zusätzlich spezielle Synchronisationsmuster
- kein Gleichstromanteil (!)

SPDIF: sync-patterns



- Signalwechsel in der ersten Zelle jedes Bits
- zusätzlich drei Synchronisationsmuster:

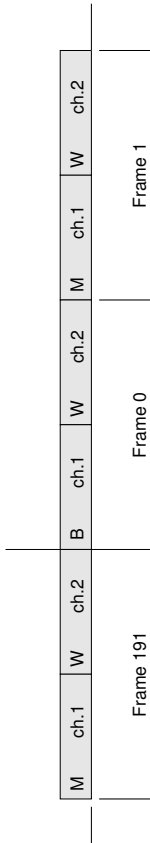
Muster	Bedeutung	last-cell 0	last-cell 1
Preamble B	left-channel data, start-of-block	11101000	00010111
Preamble M	left-channel data, inside block	11100010	00011101
Preamble W	right-channel data	11100100	00011011

SPDIF: subframe

Bits	Bedeutung
0 .. 3	preamble (sync)
4 .. 7	aux. audio data
8 .. 27	audio data, bit 27 = msb (CD uses bits 13 .. 27 (msb), 8 .. 12 are zero) (24-bit audio: bit 4 = lsb, bit 27 = msb)
28	validity, 0: valid 1: error
29	subcode data (track ID, text infos, ...)
30	channel status
31	parity of bits 4..30

- 32-bit "subframe" für jedes einzelne Datenwort
- Sync-Muster, 24-bit Audiodaten, Zusatzinfo, Parität
- nicht benutzte Audiobits sollten auf Null gesetzt werden

SPDIF: frame



- normalerweise zwei Kanäle (Stereo)
 - 192 frames (a 2 subframes) bilden einen Block
 - Sync-Muster "B" kennzeichnet Beginn des Blocks
- 192 bit channel status / block (gleiche Werte für L/R)
 384 bit subcode / block (Bedeutung nicht festgelegt)
 (1176 bit Blöcke, Sync: 16*0')

SPDIF: channel status

Bit	Bedeutung
0	0: stereo / 1: four channel transmission
1	0: digital audio / 1: non-audio (e.g. AC3)
2	0: copy prohibited / 1: copy allowed
3	0: normal / 1: pre-emphasis
4 .. 7	0 (reserved)
9 .. 15	category code: 0: 2-channel format 1: 2-channel CD format (with CD subcode)
16 .. 191	0 (reserved)

- 192 bit channel status pro Block, bisher nur einige genutzt
- Inhalt der subcode bits nicht definiert

SPDIF: SCMS

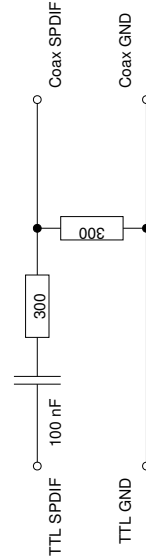
trivialer Kopierschutz SCMS:

"Serial Copy Management System"

- SPDIF-Gerät setzt das copy-protection bit zurück
- Digitalkopien direkt vom Original sind möglich
- aber keine Kopien von Kopien
- aber: in Profigeräten meistens abschaltbar . . .
- oder: externe Geräte zum Setzen des copy-bits erhältlich

SPDIF: Konverter

Umwandlung TTL-Pegel (5V) auf SPDIF-Pegel (0.5V):



- einfachste Schaltung, ohne galvanische Trennung
- weitere Docs. und Schaltungsvorschläge
<http://www.epanorama.net/documents/audio/spdif.html>
- Anwendung z.B. für SBlive out / viele CDRom-Laufwerke

ADAT



Alesis digital audio tape:

- 8-Spur Tonbandgeräte (Aufzeichnung auf Videokassetten)
- zugehörige digitale Mehrkanal-Audioschnittstelle
- weit verbreitet (im Profibereich)
- bis 8 Kanäle, 48 KHz, 24-bit Samples
- 256 bit frames @ 48 KHz: 12.288 Mb/s Bitrate (für acht Kanäle)
- optische Datenübertragung
- typisches Beispiel für selbsttaktendes Datenformat
- Datenformat effizienter als SPDIF

ADAT: Kodierung

U.S. Patent Mar. 22, 1994 Sheet 1 of 7 5,297,181

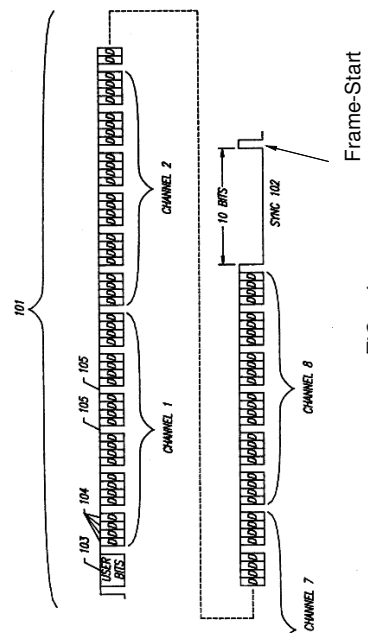
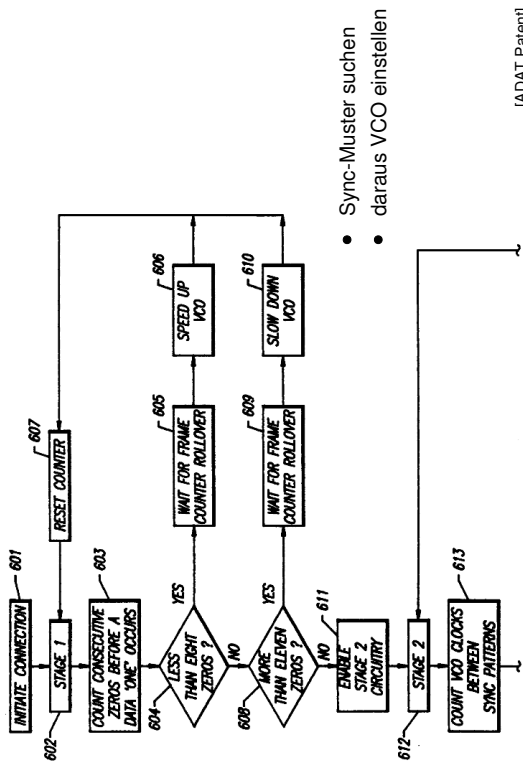


FIG. 1

- NRZI-Kodierung (1: Pegeländerung, 0: keine Änderung)
- Bitstufung: jeweils eine 1 nach vier Datenbits
- 256-bit Frame mit 8 Samples a 24 bit, stuffing, sync

ADAT: Synchronisierung



- Sync-Muster suchen
- daraus VCO einstellen

[ADAT Patent]

ADAT: Synchronisierung (2)

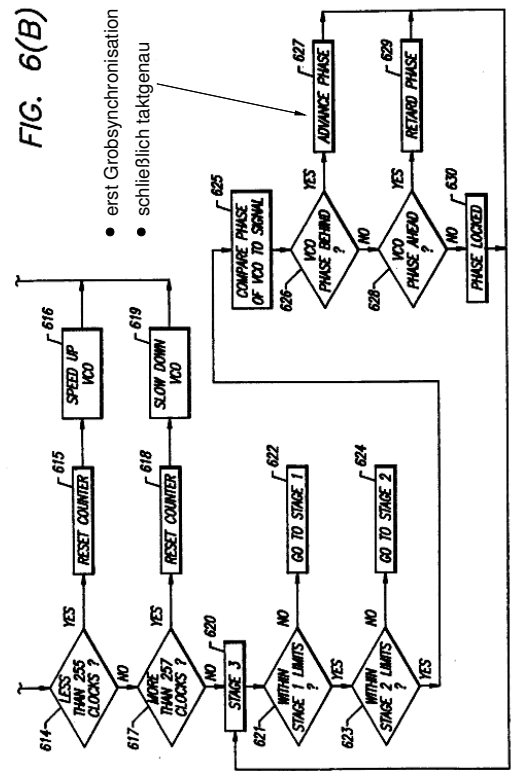


FIG. 6(B)

- erst Grobsynchronisation
- schließlich taktingenau

Perceptual Coding: Agenda

- Motivation für psychoakustische Kodierung
- Übersicht der wichtigsten Verfahren
- Literatur



MPEG1 - Audio:

- Konzept
- Encoder / Decoder
- Layer 1 / 2 / 3
- Einige Details der Algorithmen
- Beispiele zur Klangqualität
- Ausblick auf MPEG2 und MPEG4
- Ausblick auf MS-Audio (Windows Media) und RealAudio

Perceptual Coding: Motivation

perceptual coding = psychoakustische Kodierung :=

- verlustbehaftete Audiokodierung
- minimale Dateigröße bei möglichst hoher Klangqualität
- Kompression bis ca. 10:1 bei CD-Qualität: 2 bits / sample (!)
- durch Ausnutzen der Gehörwahrnehmung:
 - Ruhehearschwelle
 - Maskierungseffekte
- erfordert psychoakustische Modelle des Gehörs
- erfordert (Frequenz-) Analyse der Eingangsdaten:
 - "subband coders"
 - "transform coders"

Perceptual Coding: Verfahren

Verfahren	Methode	kHz	Kanäle	kb/s	Qualität
MPEG-1 layer 1	hybrid	32, 44.1, 48	1, 2	32 .. 448	AM / FM / (CD)
MPEG-1 layer 2	hybrid	32, 44.1, 48	1, 2	32 .. 384	AM / FM / (CD)
MPEG-1 layer 3	hybrid	32, 44.1, 48	1, 2	32 .. 320	AM / FM / CD
MPEG-2 AAC	hybrid	16 .. 48	1 .. 5.1	16 ..	AM / FM / CD
ATRAC	subband	44.1	2	256/ch	CD
Dolby AC-2	trans.	44.1	2	256/ch	CD
Dolby AC-3	trans.	44.1	1 .. 5.1	32 .. 384	CD
AT&T PAC	subband	44.1	1 .. 5.1	128/stereo	FM / CD
MS Audio ASF	?	44.1	2	10 ..	AM / FM / CD

- viele weitere Verfahren vorgeschlagen / erforscht
- detaillierte Übersicht siehe Spanias Review

Perceptual Coding: Literatur

- T. Painter, A. Spanias: Perceptual Coding of Digital Audio, Proc. IEEE 88-4, p.451ff (2000)
- T. Painter, A. Spanias: Review of Algorithms for Perceptual Coding of Audio Signals
www.eas.asu.edu/~spanias/papers.html
- D. Pan, An Overview of the MPEG/Audio Compression Algorithm, Proc. SPIE 260 (1997)
- S. Shlien, Guide to MPEG-1 Audio Standard, IEEE Trans. Broadcasting 40-4, p.206 (1994)
- Microsoft / RealAudio: msdn.microsoft.com/windowsmedia/, www.realaudio.com
- Fraunhofer IIS Dokumentation, www.iis.fhg.de
- LAME MP3 Encoder, www.sulaco.org/mp3/
- ISO Reference Software, ftp.tnt.uni-hannover.de/mpeg2/software/
- ISO/IEC Standards:
- MPEG-1, part 3: audio, ISO/IEC 11172-3 (1993)
 - MPEG-2, part 3: audio, ISO/IEC 13818-3 (1994) (Mehrkanal)
 - MPEG-4, part 3: audio
- c't, MP3-Feature (Algorithmus, Tools, Klangtest), c't 03/2000 p.122-151
- c't, MP3 vs MSAudio vs. RealAudio (mit Klangtest), c't 23/2000 p.152-161

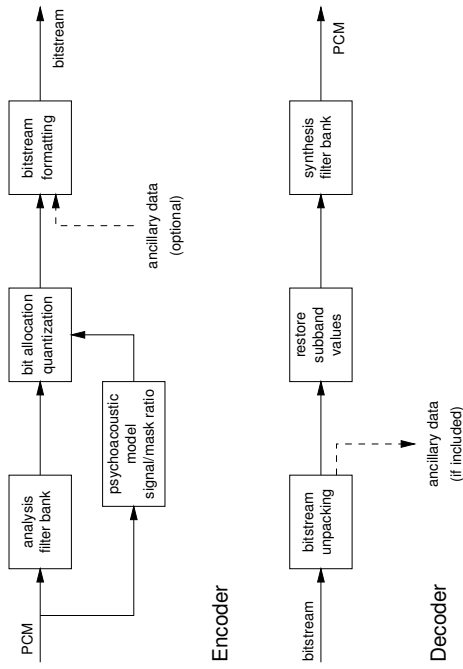
MPEG-1 Audio

- erster internationaler Standard für (HiFi-) Audio-Kompression
- entwickelt im Rahmen von MPEG-1
- bei Bitraten bis 1.5 Mb/s für Video mit Audio (Video-CD)
- verabschiedet Ende 1992 als ISO/IEC 11172-3 (1993)
- definiert das Format des Audio-Bitstroms
- und die Funktion des Decoders
- aber nicht den Encoder: Raum für spätere Verbesserungen
- Format geeignet für Sprache und Musik
- keine Quellenkodierung / keine Annahmen über Herkunft des Signals
- statt dessen psychoakustisches Modell
- Ausnutzen von Maskierungseffekten
- drei Varianten (layers), jeweils mono / stereo / joint stereo
- Samplerraten 32, 44.1, 48 KHz, Bitraten 32 .. 224 kb/s/channel

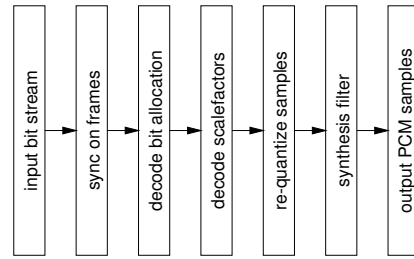
MPEG-1 Audio: Varianten

- drei Varianten, Bandbreite gesamt bis 1.5 Mb/s für Video+Audio
 - zwei psychoakustische Modelle
 - Encoder dürfen eigene psychoakustische Modelle verwenden
- Layer 1:
- einfachstes Verfahren
 - für Bitraten höher als 128 kb/s pro Kanal
 - eingesetzt für Philips DCC: 384 kb/s stereo
- Layer 2:
- mittlere Komplexität
 - für Bitraten von 128 kb/s pro Kanal
 - eingesetzt für CD-I und Video-CD
- Layer 3:
- bessere Qualität, aber deutlich komplexer
 - ab 64 kb/s pro Kanal, gute Qualität ab 128 kb/s
 - "MP3", ISDN-Übertragungen

MPEG-1: Blockschaltbild (grob)



MPEG-1: Decoder für Layer 1 und 2



- Layer 3 Decoder deutlich komplexer (siehe ISO 11172-3 S.40)

MPEG-1: Bitstrom-Header

```

audio_sequence()
{
  while (true)
  {
    frame()
  }
}

frame()
{
  header()
  error_check()
  audio_data()
  ancillary_data()
}

header()
{
  syncword      12 bits
  ID            1 bit
  layer         2 bits
  protection_bit 1 bit
  bitrate_index 4 bits
  sampling_frequency 2 bits
  padding_bit   1 bit
  private_bit   1 bit
  mode          2 bits
  mode_extension 2 bits
  copyright     1 bit
  original/home 1 bit
  emphasis     2 bits
}
    
```

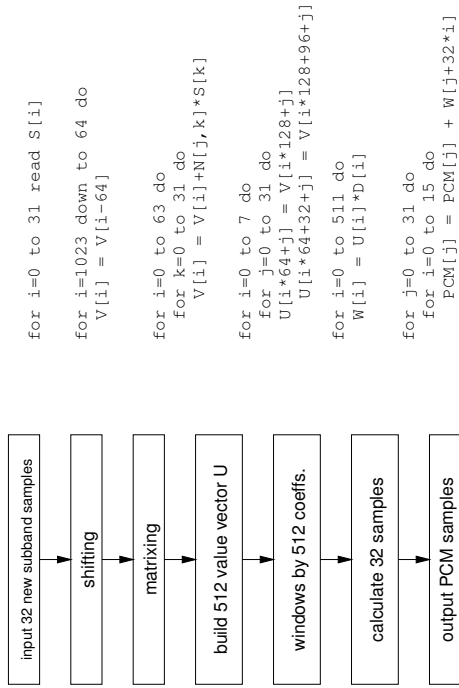
- bsbf: bit string, left bit first
- Synchronisation über Bitmuster '1111 1111 1111'

MPEG-1: Bitstrom-Daten, Layer 1

```

audio_data()
{
  if (mode==single_channel)
  {
    ...
  }
  if (mode==stereo) || (mode==dual_channel)
  {
    for (sb=0; sb<32; sb++)
    for (ch=0; ch<2; ch++)
      allocation[ch][sb] 4 bits bsmsbf
    for (sb=0; sb<32; sb++)
    for (ch=0; ch<2; ch++)
      if (allocation[ch][sb]!=0)
        scalefactor[ch][sb] 6 bits uimsbf
    for (sb=0; sb<32; sb++)
    for (ch=0; ch<2; ch++)
      if (allocation[ch][sb]!=0)
        sample[ch][sb][s] 2..15 bits uimsbf
  }
  if (mode==intensity_stereo)
  {
    ...
  }
}
    
```

MPEG-1: Synthesis Subband Filter



MPEG-1: Zahldarstellung

- möglichst kompakte Zahldarstellung gesucht
 - also mit datenabhängiger Genauigkeit
 - aber mit hohem Dynamikbereich
- => MPEG-1 Bitstrom verwendet spezielles Gleitkommaformat:
- gemeinsamer Skalenfaktor (Exponent) für eine Anzahl von Samples
 - variable Anzahl nb von Nachkommastellen
 - Auswahl der jeweiligen Stellenanzahl über Codetabellen
 - Mantisse hat nb Bits, Scalefactor aus Tabelle
 - Layer 3 zusätzlich Huffman-Codierung
- wert = scalefactor * (2**nb / (2**nb-1)) * (mantisse + 2**(-nb+1))

MPEG-1: Subband-Kodierung, Filterbank

"subband-Kodierung":

- Filterbank zerlegt das Eingangssignal
- in mehrere (gleich breite) Frequenzbänder

Idee:

- Gesamtenergie des Signals verteilt sich auf die Bänder
- und zwar meistens ungleichmäßig
- wenige Bänder mit viel Energie: präzise kodieren
- viele Bänder mit wenig Energie: mit wenig Bits kodieren
- Maskierung und Hörschwelle: kleine Koeffizienten auf Null setzen
- inverse Filterbank setzt das Signal (fast) perfekt wieder zusammen

MPEG-1: Filterbank, Layer 1 und 2

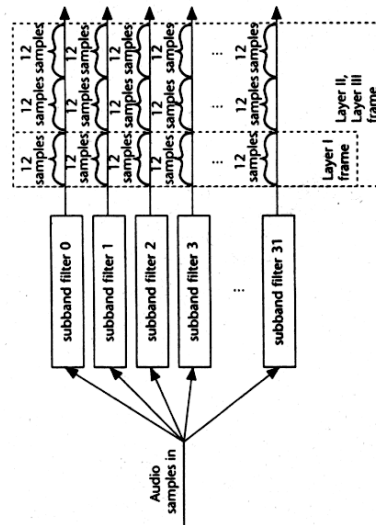


Figure 14. Grouping of subband samples for Layer I and Layer II. Note: Each subband filter produces 1 sample out for every 32 samples in.

- 32 Filterbänder, gleiche Frequenzbreite (je 20/32 KHz)
- Layer 1: 384 samples/frame input, 32x12 samples output
- Layer 2: 1152 samples/frame, 32x 3groups x 12 samples

MPEG-1: Filterbank, Layer 1 und 2

$$s_t(i) = \sum_{k=0}^7 \sum_{j=0}^7 M(i)[k] \cdot (C[k+64j] \cdot x[k+64j]) \quad (1)$$

where:

i is the subband index and ranges from 0 to 31,
 $s_t(i)$ is the filter output sample for subband i at time t , where t is an integer multiple of 32 audio sample intervals,
 $C(n)$ is one of 512 coefficients of the analysis window defined in the standard,
 $x(n)$ is an audio input sample read from a 512 sample buffer, and
 $M(i)[k] = \cos\left(\frac{(2^k+1) \cdot (k-16) \cdot \pi}{64}\right)$ are the analysis matrix coefficients.

- Berechnung der Subband-Samples $S_t(i)$
- $M(i)[k]$ ist um k bzw. j verschoben: "polyphase" Filterbank
- effizientere Berechnung via FFT möglich
- Layer 3 verwendet aufwendigere Filterbank

MPEG-1: Filterbank

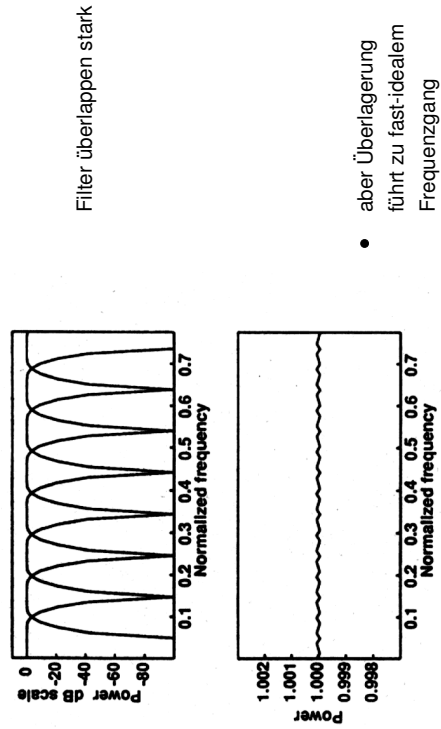


Fig. 1. Top: frequency response of the first 8 subband filters. Bottom: frequency response of the analysis filter bank. The frequency has been normalized to the Nyquist frequency 2π .

Filter überlappen stark

- aber Überlagerung führt zu fast-idealem Frequenzgang

MPEG-1: Filterbank Aliasing

- einzelne Filterbänder überlappen
- Signale nahe der Übergänge werden in zwei Bänder kodiert
- Alias-Effekte möglich
- Layer-3 verwendet nachgeschaltete MDCT-Transformation
- zur Verbesserung der Frequenzauflösung
- mit Versuch der Alias-Reduzierung

Frequenzgruppen, Bark-Skala

Tabelle 9.1: Frequenzgruppen nach Zwicker 1982

z/Bark	f_z/Hz	f_c/Hz	$\Delta f_c/\text{Hz}$	f_m/Hz
0	0	100	100	150
1	100	200	100	250
2	200	300	100	350
3	300	400	100	450
4	400	510	110	550
5	510	630	120	670
6	630	770	140	840
7	770	920	150	1000
8	920	1080	160	1170
9	1080	1270	190	1370
10	1270	1480	210	1600
11	1480	1720	240	1850
12	1720	2000	280	2150
13	2000	2320	320	2500
14	2320	2700	380	2900
15	2700	3150	450	3400
16	3150	3700	550	4000
17	3700	4400	700	4800
18	4400	5300	900	5800
19	5300	6400	1100	7000
20	6400	7700	1300	8500
21	7700	9500	1800	10500
22	9500	12000	2500	13500
23	12000	15500	3500	
24	15500			

[Zwölzer]

MPEG-1: Maskierung

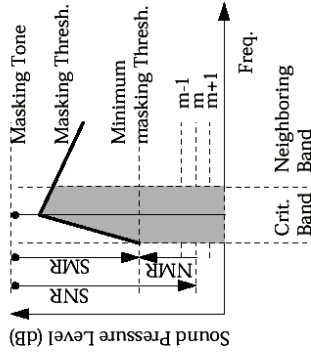
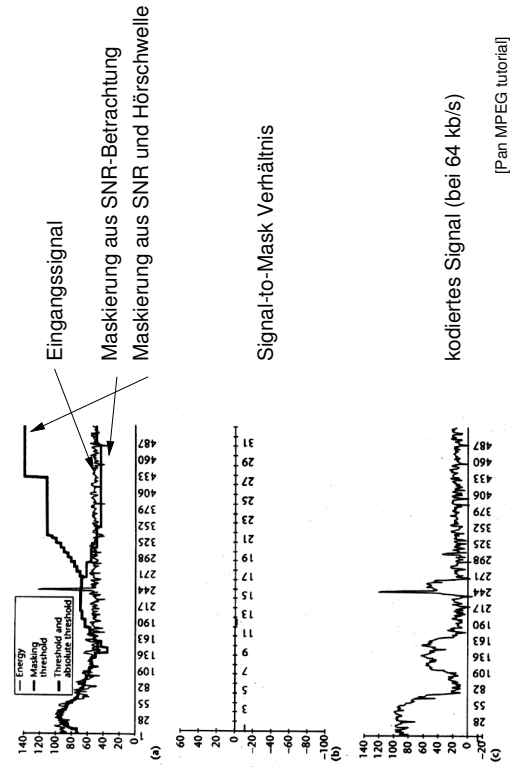


Fig. 6. Schematic Representation of Simultaneous Masking (after [30])

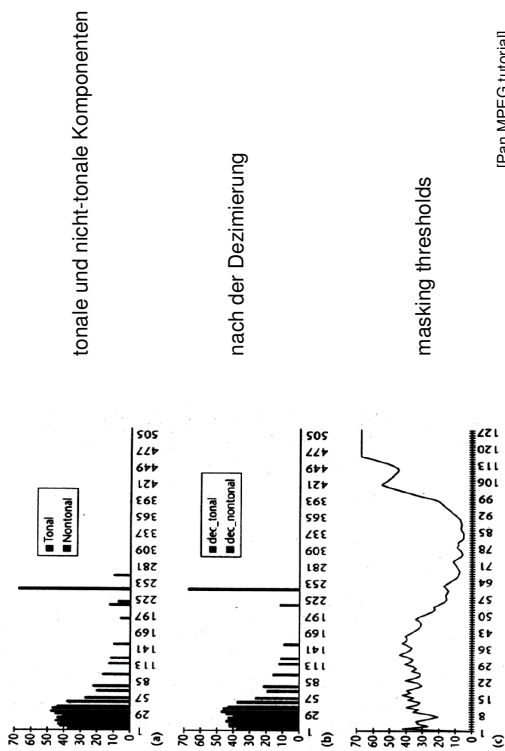
- lautester Ton im Frequenzband maskiert benachbarte
- Maskierung linear darstellbar, wenn Frequenz in Bark gemessen wird
- Töne leiser als der Masking-Threshold nicht wahrnehmbar
- SNR / SMR / NMR Verhältnisse

MPEG-1: psychoakustisches Modell 2



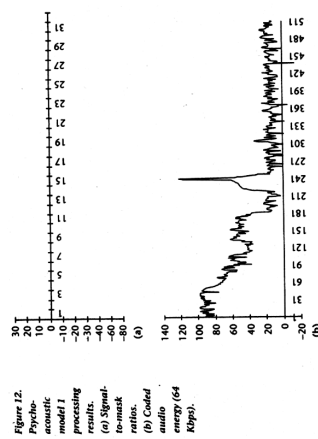
[Pan MPEG tutorial]

MPEG-1: psychoakustisches Modell 1



Digitale Audioverarbeitung | WS 2000 | 18.205

MPEG-1: psychoakustisches Modell 1



- Ergebnis der Kodierung nach Modell 1
- teilweise deutlich schlechter als Modell 2

Digitale Audioverarbeitung | WS 2000 | 18.205

MPEG-1: bit allocation

- psychoakustisches Modell liefert "signal-to-mask" Verhältnis
 - für jedes einzelne Subband
- Bit-Allocation Algorithmus:
- Berechnung von MNR und SNR (mask-noise, signal-noise-ratio)
 - wieviele Bits stehen zur Verfügung? Bitrate minus Sync. etc.
 - Aufgabe: Bitzuweisung, so daß das minimale MNR maximal
 - für die Skalierungsfaktoren
 - und die Koeffizienten aller einzelnen Subbänder
 - Details für die Layer 1,2,3 leicht unterschiedlich
 - iterativ: berechne MNR und SNR für jedes Subband
 - weise dem Subband mit der schlechtesten MNR Bits zu
 - solange noch Bits zur Verfügung stehen

Digitale Audioverarbeitung | WS 2000 | 18.205

MPEG-1: bit allocation

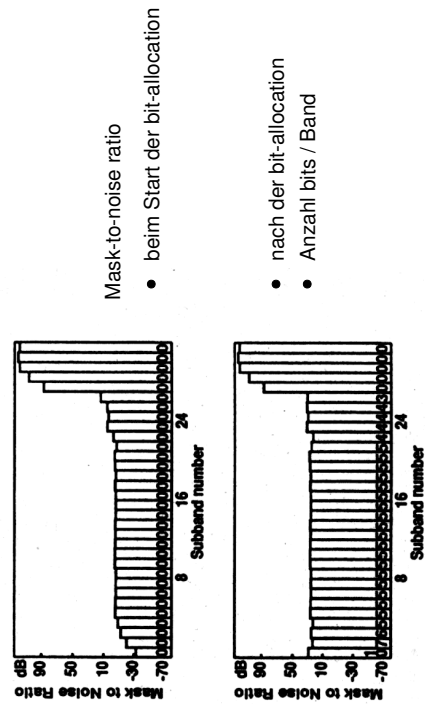


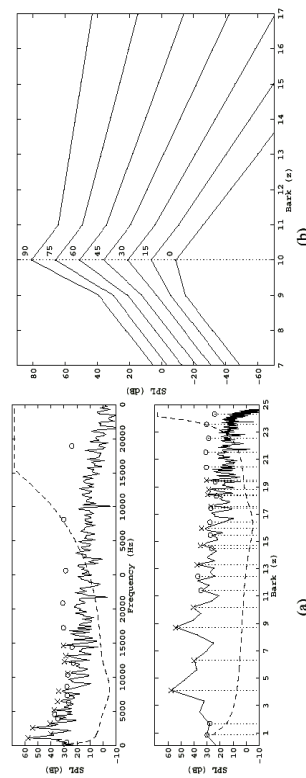
Fig. 8. Mask to noise ratio at the beginning (top) and end (bottom) of the bit allocation algorithm. The number of bits assigned to each subband sample appears at the base of each bar.

[Shlien MPEG guide]

Digitale Audioverarbeitung | WS 2000 | 18.205

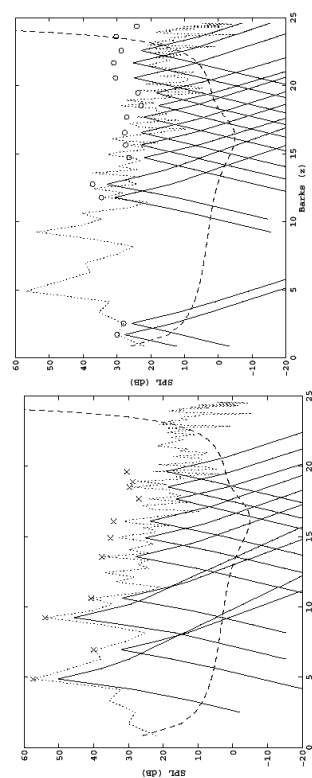
[Pan MPEG tutorial]

MPEG-1: model 1, tonal/noise maskers



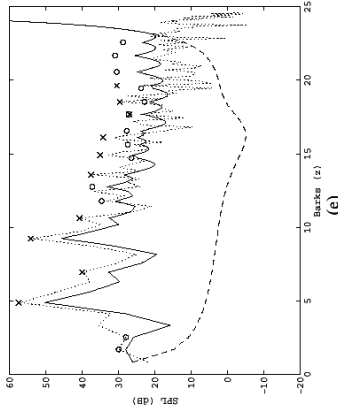
(a) Step 1: Obtain PSD, express in dB SPL. Top panel gives linear frequency scale, bottom panel gives Bark frequency scale. Absolute threshold superimposed. Step 2: Tonal maskers identified and denoted by 'X' symbol; Noise maskers identified and denoted by 'O' symbol. (b) Collection of prototype spreading functions (Eq. 28) shown with level as the parameter. These illustrate the incorporation of excitation pattern level-dependence into the model. Note that the prototype functions are defined to be piecewise linear on the Bark scale. These will be associated with maskers in steps 3, 4.

MPEG-1: model 1, masking and spreading



(c) Steps 3.4: Spreading functions are associated with each of the individual tonal maskers satisfying the rules outlined in the text. Note that the Signal-to-Mask Ratio (SMR) at the peak is close to the widely accepted value of 14.5 dB. (d) Spreading functions are associated with each of the individual noise maskers that were extracted after the tonal maskers had been eliminated from consideration, as described in the text. Note that the peak SMR is close to the widely accepted noise-masker value of 5 dB.

MPEG-1: model 1, masking and spreading



(e) Step 5: A global masking threshold is obtained by combining the individual thresholds as described in the text. The maximum of the global threshold and the absolute threshold are taken at each point in frequency to be the final global threshold. The figure clearly shows that some portions of the input spectrum require SNRs of better than 20 dB to prevent audible distortion, while other spectral regions require less than 3 dB SNR. In fact, some high-frequency portions of the signal spectrum are masked and therefore perceptually irrelevant, ultimately requiring no bits for quantization without the introduction of artifacts.

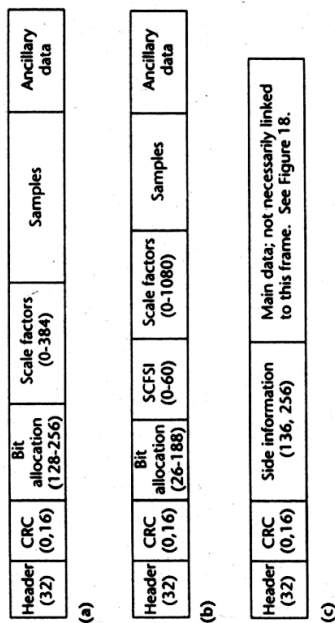
MPEG-1: Bitraten für Layer 1 .. 3

MPEG-1 Bitraten (mono/stereo/joint stereo):

index	Layer I		Layer II		Layer III	
	free	format	free	format	free	format
'0000'	32	kbit/s	32	kbit/s	32	kbit/s
'0001'	48	kbit/s	48	kbit/s	40	kbit/s
'0010'	56	kbit/s	56	kbit/s	48	kbit/s
'0011'	64	kbit/s	64	kbit/s	56	kbit/s
'0100'	80	kbit/s	80	kbit/s	64	kbit/s
'0101'	96	kbit/s	96	kbit/s	80	kbit/s
'0110'	112	kbit/s	112	kbit/s	96	kbit/s
'0111'	128	kbit/s	128	kbit/s	112	kbit/s
'1000'	160	kbit/s	160	kbit/s	128	kbit/s
'1001'	192	kbit/s	192	kbit/s	160	kbit/s
'1010'	224	kbit/s	224	kbit/s	192	kbit/s
'1011'	256	kbit/s	256	kbit/s	224	kbit/s
'1100'	320	kbit/s	320	kbit/s	256	kbit/s
'1101'	384	kbit/s	384	kbit/s	320	kbit/s
'1110'	448	kbit/s	448	kbit/s	384	kbit/s

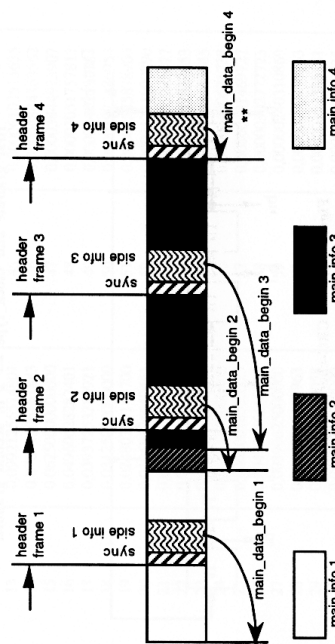
- 32 .. 448 kb/s

MPEG-1: bitstream format



- Header mit speziellem Sync-Muster, dann CRC
- Layer 3 erlaubt "borrowing" zwischen Frames

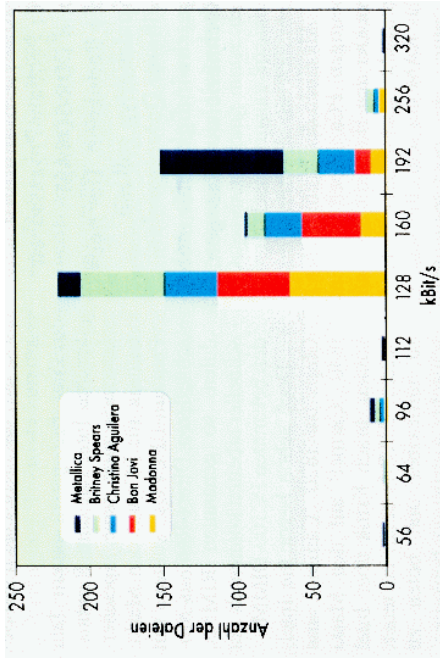
MPEG-1: bitstream format



**) main_data_begin 4 == 0: This signifies that main data starts directly after the side information for frame 4. This is the lower limit for main_data_begin; main_data cannot start later than this point. Note that data bytes used by "sync" and "side info" are not counted by the main_data_begin pointer.
 Note: "info" means information

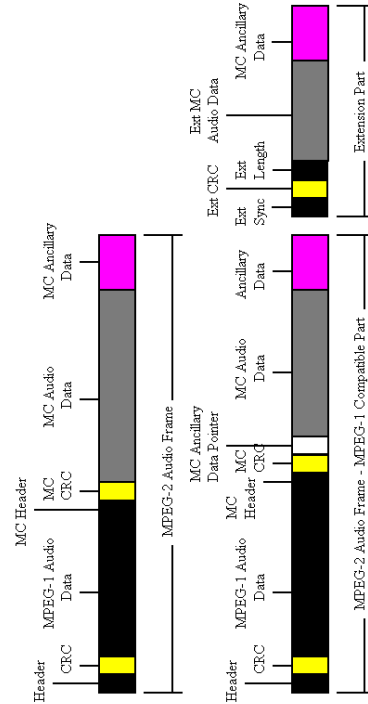
Figure A.7.b -- Layer III bitstream organization with peak demand at main info 3 and small demand at main info 2.

MPEG-1: gängige Bitraten



- Histogramm der Bitraten (MP3)
- Suche per Napster, aus [c't 2/2000]

MPEG-2: multichannel bitstream



- MPEG-1 compatibles Stereosignal, plus Mehrkanal-Erweiterung

Figure A.2 Structure of the ISO 13818-3 Layer II multichannel extension, using the ISO/IEC 11172-3 compatible bit stream as well as the extension bit stream

MPEG-1: pre-echos

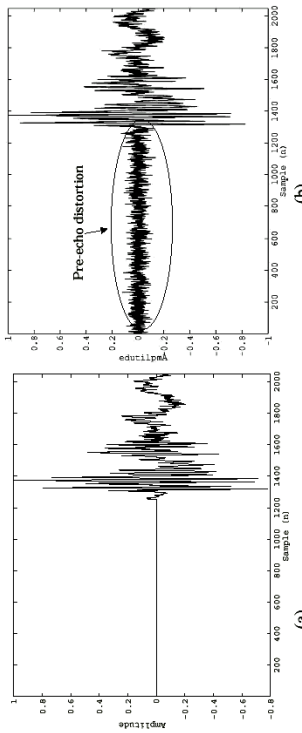
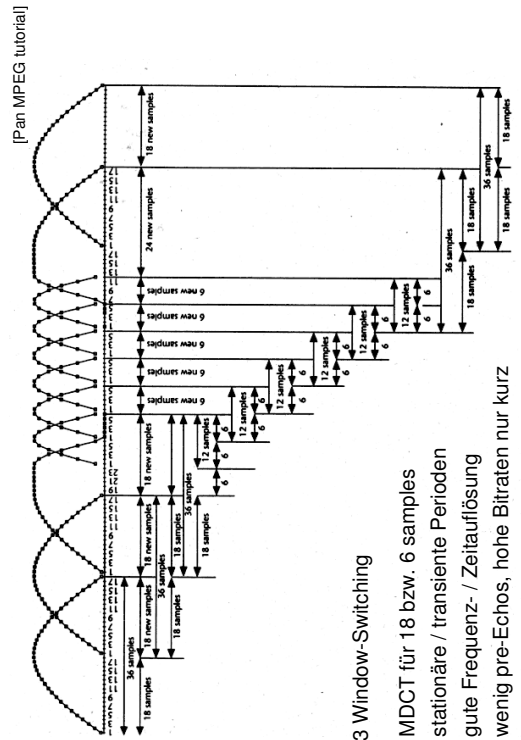


Fig. 14. Pre-Echo Example: (a) Unencoded Castanets. (b) Transform Coded Castanets, 2048-Point Block Size [Painter Spanias]

- Problem mit plötzlichen lauten Signalen nach einer Blockgrenze
 - vorheriger Block hat sehr wenig Bits kodiert
 - Rauschen des lauten Signals wird nicht mehr maskiert
- => Situation erkennen und behandeln (kürzere MDCT Blöcke)

MPEG-1: window-switching



Layer 3 Window-Switching

- MDCT für 18 bzw. 6 samples
- stationäre / transiente Perioden
- gute Frequenz- / Zeitauflösung
- wenig pre-Echos, hohe Bitraten nur kurz

MPEG-1: temporal noise-shaping

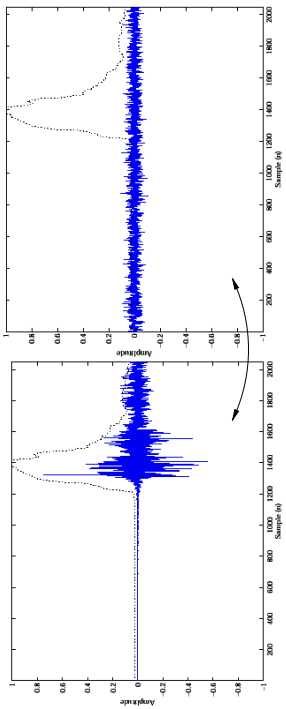


Fig. 17. Temporal Noise Shaping Example Showing Quantization Noise and the Input Signal Energy Envelope for Castanets: (a) Without TNS, and (b) With TNS [Painter Spanias, a) b) vertauscht]

weitere Verbesserung durch TNS:

- Verlagerung des Rauschens in den Bereich mit höchster Maskierung
- wird für MPEG-2 AAC eingesetzt

MPEG-2: BC und NBC (AAC)

"backwards compatible":

- ähnliches Format wie MPEG-1 layer 3
- aber mit 3/2-Mehrkanal-Erweiterung
- abwärtskompatibel (für die 2/0 Stereosignale)
- ungeeignet für 3/2 mit Bitraten unter 640 kb/s

"non backwards compatible" / "advanced audio coding"

- gleiches Konzept wie MPEG-1 layer 3
- aber viele Verbesserungen, mit Ziel geringerer Bitrate
- viele neue Details, bessere Filterbank, bessere Zeitauflösung
- temporal noise shaping, . . .
- zusätzliche Bitraten bis herunter zu 16 kb/s/channel

MPEG-2: AAC Eigenschaften

MPEG-2 AAC (ISO 13818-7, 1997):

- Mehrkanal-Audio (3/2) bei Bitraten bis 384 kb/s
- transparente Kodierung (Hörtests) ab 320 kb/s
- mehrere "Profile": main / low / scalable sample rate profile
- bessere Filterbank als MP3:
- Frequenzauflösung 23 Hz (75Hz), Zeitauflösung 2.6 ms
- keine Hybrid-Filterbank, sondern nur MDCT
- MDCT window shape switching (stationary / transient signals)
- time prediction of transform coefficients
- more efficient Huffman code tables
- temporal noise shaping
- scalability and complexity management tools
- eingesetzt u.a. für DVD und Liquid Audio

MPEG-2: AAC-Decoder Performance

"revised report on complexity of MPEG-2 AAC tools2"
 "ISO/IEC JTC1/SC229/WD11/N2957 MPEG99"

welche Hardware-Anforderungen für AAC-Dekodierung?!

Decoder	Complexity
2-channel MP software decoder	40% Pentium/133
2-channel low complexity sw decoder	25% Pentium/133
5-channel MP hardware decoder	90 mm² 0.5 µm CMOS
Huffman decode	13.3 %
inv. quantization, scaling	1.7 %
temporal noise shaping	6.6 %
prediction	43.2 %
IMDCT	19.4 %
coupling channel	15.5 %

MPEG-2: und Nachfolger

- entwickelt für geringe Bitraten
- Hörtests bescheinigen (sehr) gute Qualität
- auch für kritisches Audio-Material
- aber nicht unbedingt besser als MP3
- siehe MPEG-2 quality report
- Hörtests / Vergleich mehrerer Algorithmen
- MPEG-4 (ISO/IEC 14496, 1998) enthält MPEG-2
- zusätzlich "objekt-orientierte" Algorithmen
- für Bitraten von 64 kb/s bis hinunter zu 200 b/s (l)
- für Sprachkodierung / -synthese
- für Musikkodierung / -synthese (a la MIDI)

MPEG-1: LAME

"Lame ain't an MPEG encoder"

- MP3-Encoder unter GPL Lizenz, im Quellcode verfügbar
- ursprünglich als Patch gegen den ISO Referenzcode
- derzeit als völlig neuentwickeltes Programm
- Decoder basiert auf mpeglib (www.mpg123.de)
- mit eigenem psychoakustischem Modell
- Qualität vergleichbar mit dem FhG-Encoder
- ideal als Grundlage für Eigenentwicklungen
- unterstützt MPEG-1 layer 1 bis 3
- recht schnelle Kodierung
- www.sulaco.com/mp3 (aktuell ist Lame 3.87)

LAME: psychoakustisches Modell

```

/* PsyModel.c (c) 1999 Mark Taylor
 * Copyright (c) 1999 Mark Taylor
 * Compute FFTs mid and side channel is derived from L & R
 * Compute energies
 * Compute unpredictability of first six spectral lines
 * Compute unpredictability of next 200 spectral lines
 * Calculate the energy and the unpredictability in the threshold
 * calculation partitions
 * convolve the partitioned energy and unpredictability
 * with the spreading function, sp_[L|R][k]
 * calculate the tonality of each threshold calculation partition
 * calculate the energy in each threshold calculation partition
 * tonality = 0; use NMT (lots of masking)
 * tonality = 1; use TMN (little masking)
 * at this point, tbb represents the amount the spreading function
 * will be reduced. The smaller the value, the less masking.
 * long block pre-echo control for both short and long blocks
 * threshold calculation for short blocks
 * compute M/S thresholds from Johnston & Ferreira 1992 ICASSP paper
 * determine final block type
 * compute side energy / (side+mid)_energy
 * 0 = no energy in side channel
 * .5 = half of total energy in side channel
 * Read long block data for converting threshold calculation
 * Read short block data for converting threshold calculation
 * compute which partition bands are in which scalefactor bands
 * compute bark value and ATH of each critical band
 * Now compute the spreading function, s[j][i], the value of the spread-
 * ing function, centered at band j, for band i, store for later use
 * / setup stereo demasking thresholds
  
```

LAME: Status, Performance

- Echtzeit-Kodierung ist möglich
 - auch bei hohen Bitraten / guter Qualität
- ```

vs. Echtzeit
lame --preset studio 256 kb/s 3X
lame --preset cd 192 kb/s 3X
lame --preset fm 112 kb/s 3X
auf Celeron/466

```
- psychoakustisches Modell liegt im Quelltext vor
  - weitere Verbesserungen vorgeschlagen und geplant
  - aber massive rechtliche Probleme:
  - fast alle MPEG-1/2 Verfahren patentiert und lizenzpflichtig
  - siehe MPEG-2 AAC Lizenzvertrag (PDF)

## LAME: Presets

Presets are some shortcuts for common settings.  
They can be combined with -v if you want VBR MP3s.

```

=====
phone phont+ lw sw fm voice radio tape hifi cd studio
=====
--resample 8 11 11 11 32 24
--highpass 0.125 0.1 4 4 15 12 15 18 18
--lowpass 3.4 4 4 0 0 0 0 0 0.9 0.9
--lowpass-width yes yes no no yes no no no no no
--noshort -mm -mm -mm -mm -mm -mm -mm -mm -h -h
-b 16 24 24 24 112 56 128 128 160 192 256
-- PLUS WITH -v -----
-v 6 4 3 3 3 4 3 3 2 1 0
-b 8 16 16 16 80 40 96 112 128 160
-B 24 32 56 56 256 112 256 256 320 320 320
=====
EXAMPLES:
a) --preset fm
 equal to: -mj -b112 --resample 32 --lowpass 15 --lowpass-width 0
b) -v --preset studio
 equals to: -h -ms -v0 -b160 -B320

```

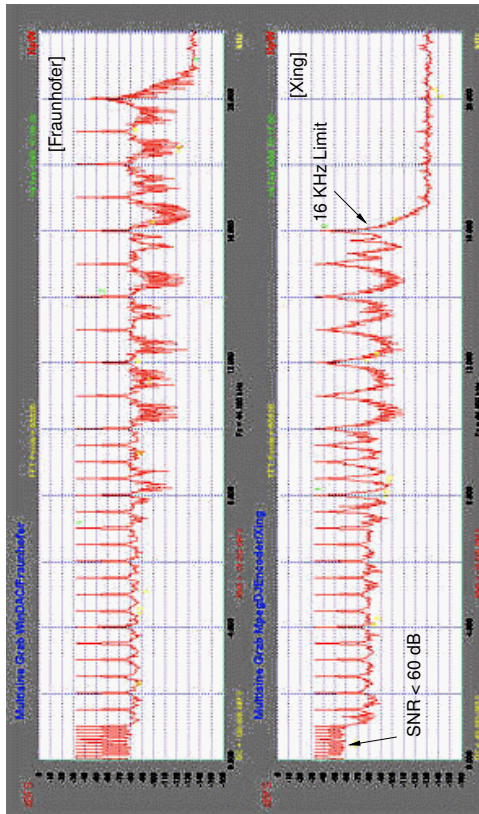
- feste Bitraten von 16 kb/s (mono) bis 256 kb/s (stereo)
- -B setzt maximal zulässigen Wert bei variabler Bitrate
- -V 0 für beste Qualität, -V 9 für minimale Dateigröße

## Klangvergleich

|            | MP3 | WMA | RA8 | ePAC | AAC | MP+ |
|------------|-----|-----|-----|------|-----|-----|
| Modem      | 0   | ++  | +   | ++   | -   | X   |
| Rock       | +   | +   | +   | +    | 0   | ++  |
|            | 160 | 160 | 185 | 160  | 256 | 184 |
| Pop        | +   | ++  | +   | ++   | +   | ++  |
|            | 160 | 128 | 185 | 128  | 192 | 171 |
| Klassik    | +   | +   | ++  | +    | +   | ++  |
|            | 160 | 160 | 136 | 160  | 190 | 158 |
| Stresstest | +   | --  | 0   | --   | ++  | ++  |
|            | 320 | 192 | 360 | 160  | 320 | 296 |

- Vergleichstest in c't 2/2000 p.152
- Klangqualität bei 128/138 kb/s / Bitrate für CD-nahe Qualität
- MP+ ist Eigenentwicklung der Autoren, "audiophil"

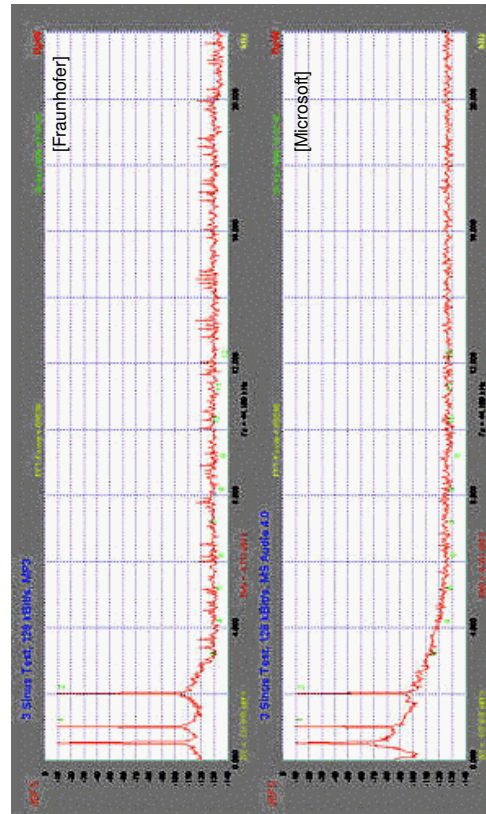
**MP3: zwei Encoder ...**



- Eingangssignal mit 44 Sinustönen ...

[ct 21/98 242]

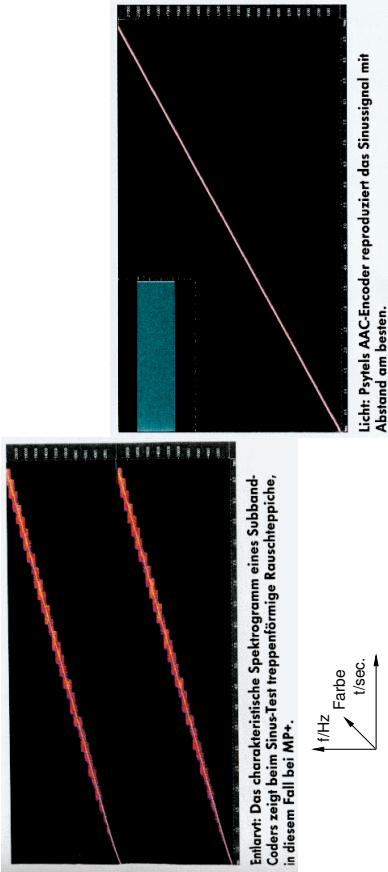
**MP3: vs. MS Audio 4.0**



- neuere Versionen von WMA sind besser

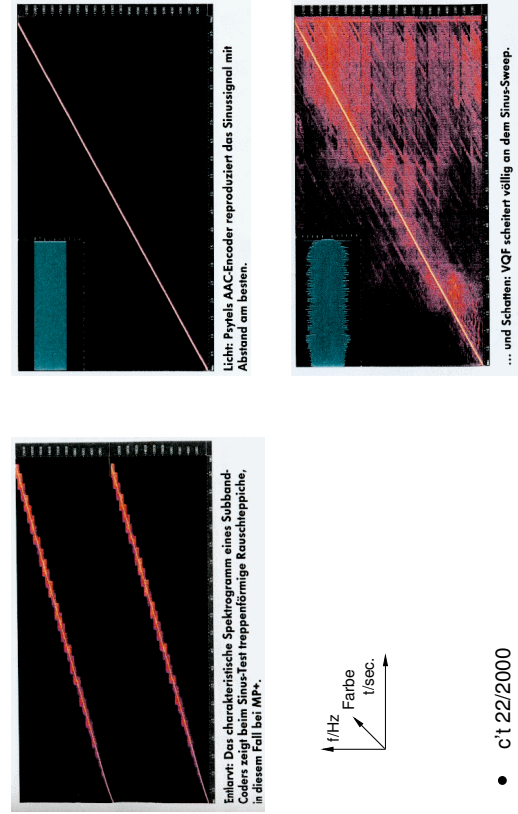
[ct 21/98 242]

**MPEG-I: Sinus-Sweep**



- Subband-Kodierung zerlegt das Signal in Frequenzbereiche
- Frequenz-Sweep besonders schwer zu kodieren
- Test für MP3, AAC, MP+, VQF in ct 22/2000, 152

**Sinus-Sweep: MP+, AAC, VCQ**



- ct 22/2000

### Streaming: Agenda

- "Definition"
- Übersicht
- Protokolle: RTP, RTCP, RTSP, RSVP
- Microsoft ASF und WMA
- RealAudio, Apple Quicktime, LiquidAudio
- Napster
- Gnutella
- später:
  - SDMI, Verschlüsselung, Wasserzeichen
  - Mediensuche

### Streaming: Literatur

troiz Bedeutung des Themas:  
 bisher keine Lehrbücher, Tutorials  
 => statt dessen nur Weblinks / Hersteller-Whitepaper / ...

Internet RFCs, insbesondere RFC1889 für RTP und RTCP

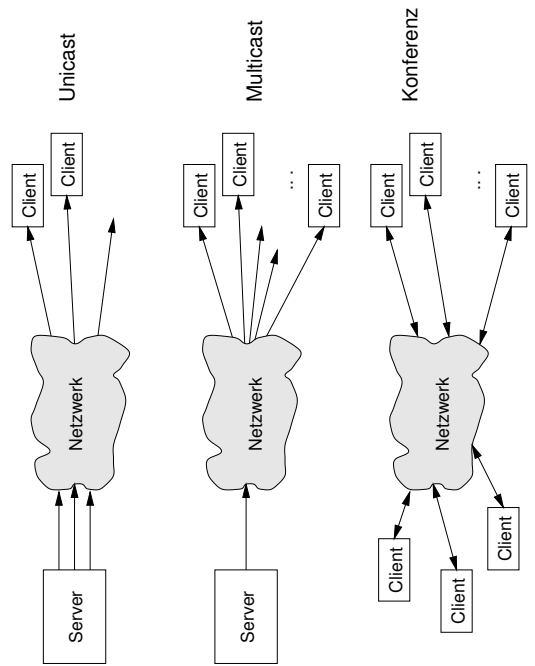
[www.microsoft.com/windowsmedia](http://www.microsoft.com/windowsmedia)  
[www.realaudio.com](http://www.realaudio.com)  
[www.apple.com/quicktime](http://www.apple.com/quicktime)  
[www.napster.com](http://www.napster.com)  
[gnutella.wego.com](http://gnutella.wego.com)  
[www.liquidaudio.com](http://www.liquidaudio.com)

### Streaming: Definition

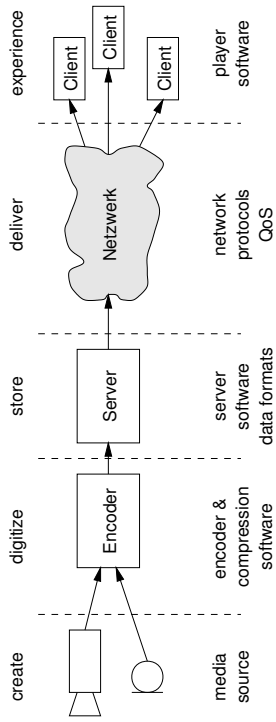
"Streaming Media / Audio" :=

- Echtzeit-Übertragung von Medien, hier: Audiodaten
- von einem Server zu einem / vielen Clients
- ohne vorheriges Laden der gesamten Daten
- Beschränkung auf Bitrate des Kanals, mit Schwankungen
- erfordert Fehlertoleranzmechanismen
- Tradeoff Qualität / Robustheit / Bitrate
- Audio auch standalone
- aber meistens Video / Animationen mit synchronem Audio
- großes Marktpotential (z.B. Internet-Radio, Video on demand, ...)
- Microsoft WMA, RealAudio, Apple Quicktime, Liquid Audio, ...
- Mediensuche / Tauschbörsen (Napster) / Copyright-Fragen

### Streaming: Szenarien



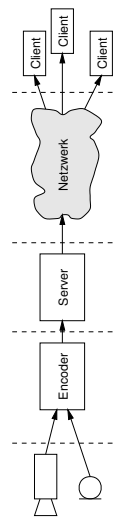
### Streaming: System



funktionierendes System erfordert alle Komponenten:

- alle Anbieter liefern "integrierte Lösung"
- Encoder + Server + Management + Player (+ Verschlüsselung)
- Microsoft, RealNetworks, Apple, ...
- Live-Übertragung erfordert auch Echtzeit-Encoder

### Streaming: Vor- und Nachteile



Vorteile:

- keine Wartezeit während der Dateiübertragung
- sofortiges Reinhören ("tune in")
- Live-Übertragungen
- kein Speicherplatz auf dem Client benötigt

Nachteile:

- durch verfügbare Bandbreite begrenzt
- deshalb oft noch schlechte Qualität
- spezielle Server und Software erforderlich

### Streaming: Bandbreiten

"streaming" ist stark von verfügbarer Bandbreite abhängig:

|               |            |      |
|---------------|------------|------|
| Modem         | 28 .. 56   | Kb/s |
| ISDN          | 64         | Kb/s |
| DSL           | 128 .. 768 | Kb/s |
| Intranet      | 10 .. 100  | Mb/s |
| MPEG-2 (DVD)  | 4 .. 9     | Mb/s |
| MPEG-1, CDDA  | 1.5        | Mb/s |
| MPEG-4 (DivX) | 1.5        | Mb/s |
| MP3 Audio     | 64 .. 320  | Kb/s |
| RealAudio 8   | 10 .. 160  | Kb/s |

(bis ca. 4 Mb/s)

- => selbst MP3 nicht per Modem streaming-fähig
- => Bildtelefonie (H.263 QCIF 15fps) erfordert mindestens ISDN

### Streaming: QoS

"Quality of Service" ?!

- Internet ist "best effort" Netzwerk
- selbst innerhalb von Firmen-Intranets keine Garantien
- ATM hat sich nicht durchgesetzt

=> keine Bandbreite reservierbar

=> Fehlertoleranzmechanismen notwendig:

- forward error correction
- Dienstgütemanagement
- adaptive Verfahren
- Forschungsschwerpunkt bei TKRN (Prof. Wolfinger)
- insbesondere QoS bei Videokommunikation

### Streaming: Marktbedeutung

- digitale Techniken kurz vor dem Durchbruch
- mit DSL-Modem ist Video-on-demand möglich
- => enormes Marktvolumen
- aktiver Kampf um Marktanteile
- => proprietäre, nicht dokumentierte Protokolle
- => Standards sind zu langsam (!)
- Marktführer: Microsoft, RealNetworks, Apple
- aber Revolutionen jederzeit möglich, siehe Napster
- oft keinerlei Dokumentation verfügbar
- Testberichte / Vergleichstests fragwürdig
- siehe "NSTL" Tests: MS vs. RealAudio

### Streaming: Internet

- Multimedia-Übertragung im Internet ("IP"):
- hohe Bandbreiten erforderlich (s.o.)
  - für Unicast / Multicast-Übertragungen
  - nur Paketvermittlung, keine Punkt-zu-Punkt-Verbindungen
  - Echtzeitanforderungen: z.B. Latenz < 250ms für Telephonie
  - => "klassische" Protokolle (ftp) für Streaming ungeeignet
  - => neue Protokolle notwendig
  - verlorene Daten (dropped frames) oft tolerierbar
  - verlorene Pakete meistens bei überlastetem Netzwerk
  - wiederholte Übertragung verschlimmert das Problem
  - => auf IP/UDP aufsetzen, nicht auf TCP

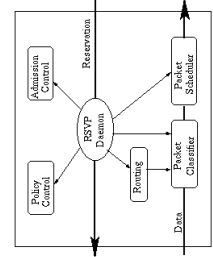
### Streaming: Internet-Protokolle

- neue Protokolle für Multimedia-Daten:
- Pakete klassifizieren und priorisieren
  - Reservierung von Resources / Bandbreite / Routes
  - Anpassung an Netzauslastung
  - RSVP resource reservation protocol RFC2205..2209
  - RTP real time transport protocol RFC1889, 1890
  - RTCP real time control protocol
  - RTSP real time streaming protocol
  - proprietäre Protokolle (Aufsetzen auf IP/UDP)

### RSVP

"resource ReSerVation Protocol" : RFC 2205..RFC2209 (approved 1997)

- Internet "network control protocol"
- Empfänger beantragt garantierte Bandbreite (QoS) für eine Punkt-zu-Punkt Verbindung
- bei allen beteiligten Routern
- realisiert über verändertes (priorisiertes) Packet-Scheduling
- RSVP überträgt keine Nutzdaten
- rsvpD Dämon-Prozess auf jedem Router
- "admission control" Ressourcen übrig ?
- "policy control" Zugriffsrechte ?
- jeder Router hat "Veto-Recht"







### RTCP: Receiver Report

```

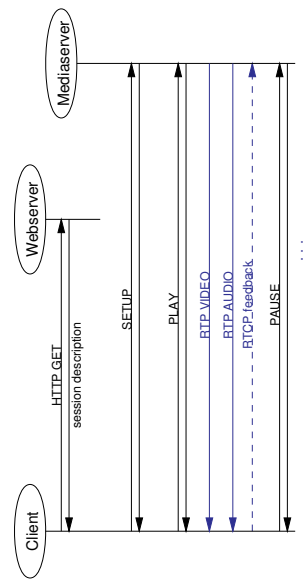
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|V=2|P| RC | PT=RR=201 | length | header |
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSRC of packet sender |
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSRC_1 (SSRC of first source) | report |
+-----+-----+-----+-----+-----+-----+-----+-----+
| fraction lost | cumulative number of packets lost |
+-----+-----+-----+-----+-----+-----+-----+-----+
| extended highest sequence number received |
+-----+-----+-----+-----+-----+-----+-----+-----+
| interarrival jitter |
+-----+-----+-----+-----+-----+-----+-----+-----+
| last SR (LSR) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| delay since last SR (DLSR) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSRC_2 (SSRC of second source) | report |
+-----+-----+-----+-----+-----+-----+-----+-----+
: :
+-----+-----+-----+-----+-----+-----+-----+-----+
| profile-specific extensions |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

### RTSP

"real time streaming protocol" := "Internet Videorecorder"

- Steuerung von Medienübertragung: play, ff, rewind, ...
- Medien-URLs: rtsp://media.example.com:554/twister/
- RTP/RTCP/RSPV für Datentransfer



### RTSP: URLs

- siehe RTSP-Präsentation (H. Schulzrinne)

### Microsoft: WindowsMedia

- seit ca. Q4/98 enormes Engagement von Microsoft: [www.microsoft.com/windowsmedia/](http://www.microsoft.com/windowsmedia/)
- Windows Media Player 6.4 / 7 / 8 beta
  - eigenes Audioformat
  - eigenes Videoformat
  - Formate nicht mehr dokumentiert
  - Verschlüsselung und Wasserzeichen
  - aber auch Unterstützung von Standards
- Windows Media Tools
  - Windows Media Encoder
  - Presentation Broadcast (Powerpoint+Audio)
  - Windows Media Rights Manager

## WindowsMedia: Ziele, Features

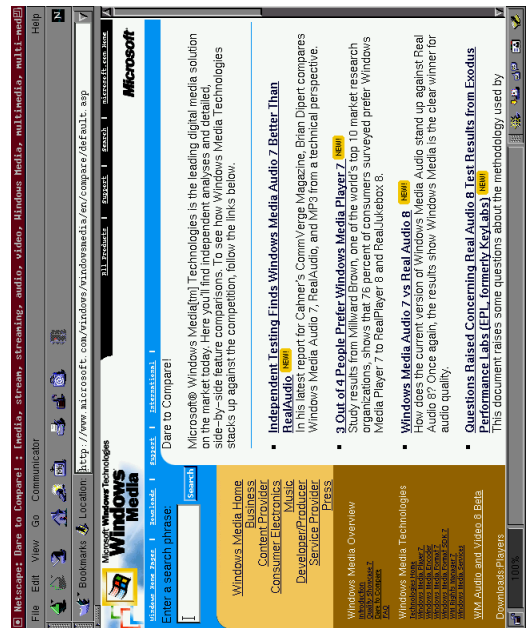
"deliver digital media, such as songs and videos, through the Internet in a protected and secure manner. [...] Provides tools for protecting digital files so you distribute them and maintain copyright protection. [...] Solutions are flexible enough to be easily adapted to your current business model or to create new ones."

Microsoft Media Rights Manager 7 SDK

Komponenten des WindowsMedia Konzepts:

- Windows Media Player 7
- Windows Media Tools
- Windows Media Rights Manager 7
- Player auch für nicht-Windows Plattformen verfügbar derzeit: Macintosh, Solaris (!), Pocket-PC, "palm-size PC"

## Microsoft WindowsMedia: Website



## WindowsMedia: Media Player 7



Skins :-)

wesentliche Veränderungen seit Player 6.x auch unter der Oberfläche:

- online feedback über Mediennutzung
- unterstützt Windows Media Rights Manager
- z.B. "counted operations", "expiration", ...

## AVI: Audio Video Interleaved



normal / interleaved

- RIFF-Dateiformat für Multimedia / Videos
- eingeführt mit Video for Windows (Win 3.11)
- definiert ca. 20 Chunk-Typen
- List-Chunk erlaubt "verwobene" Daten
  - Audiodaten, Videoframes (BMP), Audiodaten, ...
  - Datei muß vor Abspielen nicht voll geladen werden
- Spezifikation in Windows API, Übersicht in c't 94/11 S.327
- mittlerweile von ASF abgelöst (s.u.)

### AVI: Beispiel

```

(0x00000000) ID:<RIFF> Size:0x000fe964 RIFF: AVI
Form Type = <AVI >
ID:<LIST> Size:0x000007d4 Header
List Type = <hdr1> [c't11/94 327ff]
ID:<avib> Size:0x00000038
ID:<LIST> Size:0x00000474
List Type = <str1>
ID:<stsh> Size:0x00000038
ID:<strf> Size:0x00000428
ID:<LIST> Size:0x0000005c
List Type = <str1>
ID:<stsh> Size:0x00000038
ID:<strf> Size:0x00000010
ID:<svedt> Size:0x00000008
ID:<JUNK> Size:0x00000298
ID:<LIST> Size:0x000fd184
List Type = <movi>
ID:<LIST> Size:0x000005b0
List Type = <rec >
ID:<O1wb> Size:0x000005a4
/* einige Male nur Sound */
ID:<LIST> Size:0x000019da
List Type = <rec >
ID:<O0dc> Size:0x00001422
ID:<O1wb> Size:0x000005a4
/* diverse Male Sound und Bild */
ID:<LIST> Size:0x00002d72
List Type = <rec >
ID:<O0dc> Size:0x00002d66
/* einige Male nur Bild */
ID:<O1wb> Size:0x00000f00
Index

```

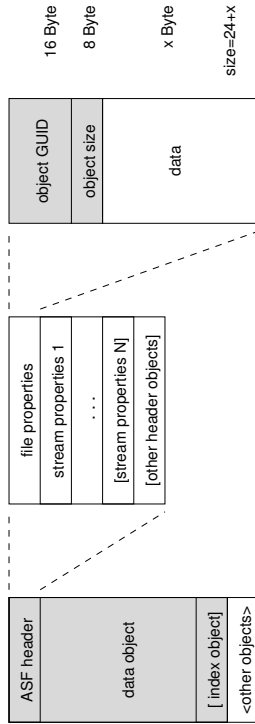
### ASF: Advanced Streaming Format

"... an extensible file format designed to store synchronized multimedia data. It supports data delivery over a wide variety of networks and protocols, while still proving suitable for local playback. The explicit goal of ASF is to provide a basis for industry-wide multimedia interoperability, with ASF being adopted by all major streaming solution providers."

[Microsoft, Real Networks 1998]

- RIFF-ähnliche Struktur: header, index, interleaved data
- Chunks (= "objects") per GUID gekennzeichnet
  - bei Bedarf Registrierung der GUIDs bei Microsoft
  - erlaubt z.B. Codec-Download
  - auch der Player identifiziert sich über seine GUID
  - sehr feine Copyright / Nutzungskontrolle
- siehe ASF-Spezifikation (Version 1.0, 26.02.1998)
- aber aktuelle Version (WMA) von MS nicht mehr dokumentiert

### ASF: Object



- Datei besteht aus ASF objects (statt RIFF chunks)
- oberste Ebene: header / data / [Index] / [more objects]
- weitere Ebenen durch Einteilen des data-object
- insbesondere: mehrere Datenströme innerhalb des data-object
- Header-Object bei Bedarf wiederholt senden
- Größenfeld erlaubt Objekte bis zu 2\*\*64 Byte

### ASF: Audio object

| Field Name             | Type | #bits |
|------------------------|------|-------|
| Codec ID               | GUID | 128   |
| Error Concealment Type | GUID | 128   |
| Bits per Sample        | UINT | 32    |
| Samples per Second     | UINT | 32    |
| Avg. Frame Size        | UINT | 32    |
| Max. Frame Size        | UINT | 32    |
| Samples per Frame      | UINT | 32    |
| Flags                  | UINT | 16+16 |
| Number of Channels     | UINT | 16    |
| ...                    | ...  | ?     |

- vordefinierte "media types" für: Audio / Video / Image / MIDI / Text / Timecode / ...

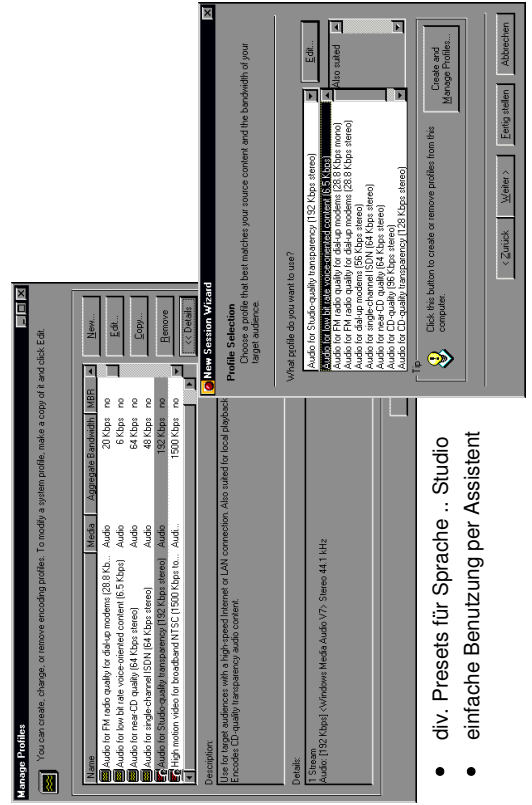
## WindowsMedia: Media Encoder

Windows Media Encoder:

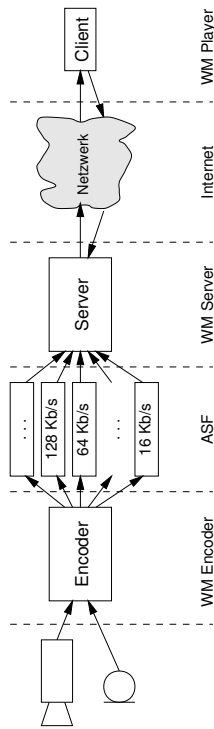
- Basiskomponente für Windows-Media Authoring
- liest diverse Formate (WAV, MP3, ..., MPEG-1, ...)
- und schreibt WMA
- erlaubt diverse Text-Infos (Autor, Copyright, Titelname, ...)
- diverse "profiles" mit Voreinstellungen
- Datenformat nicht mehr dokumentiert
- erweitertes ASF, plus Verschlüsselung
- Encoder läuft auch als Server
- bis 50 Unicast-Clients

Digitale Audioverarbeitung | WS 2000 | 18.205

## WindowsMedia: Audio Profiles, Wizard



## WindowsMedia: Intelligent Streaming



- Encoder erzeugt (optional) spezielle ASF-Datei:
  - mehrere parallele Datenströme
  - mit verschiedenen Bitraten
  - entsprechende Qualitätsstufen
- Player sendet Feedback über Bandbreite und Paketverluste
- Server sendet nur den Datenstrom der max. möglichen Qualität

## WindowsMedia: Security

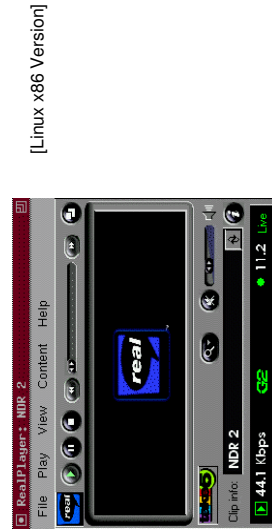
- mögliche Restriktionen für WMA/WMV-Dateien:
  - play on PC
  - counted play
  - start date
  - expire date
  - burn to Audio CD
  - counted CD burn
  - transfer to portable device
  - ... and many more
- Player überprüft vorhandene Lizenzen
- oder versucht, neue Lizenz zu erhalten:
  - "might issue a licence silently, so the consumer is unaware of the process"
  - "... supports pre-delivered licences"

## RealAudio

Real Networks : Marktführer für "streaming media"

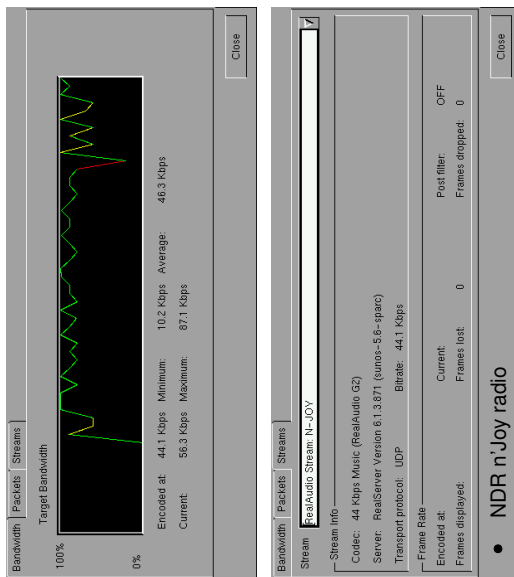
- RealSystem G2: Audio- und Videostreaming
- auch für Modemverbindungen
- mit Bitraten < 28.8 kbps
- Klangqualität entsprechend schlecht (auch laut Vergleichstests)
- Player verfügbar für viele Plattformen
- z.B. als Plugin im Standardumfang von Netscape 4.x enthalten
- RealAudio8: seit 23.10.2000
- basiert auf Sony ATRAC3 (wie MiniDisc)
- CD-nahe Qualität ab 64 kbps
- Whitepaper unter [www.realnetworks.com/realsystem/tech\\_overview.html](http://www.realnetworks.com/realsystem/tech_overview.html)

## RealAudio: RealPlayer



- unterstützt RealAudio G2 und RealAudio 8
- aber neues Format noch wenig verbreitet

## RealAudio: Statistik



Digitale Audioverarbeitung | WS 2000 | 18.205

## RealAudio: RealServer

- |                         |                                                                                        |
|-------------------------|----------------------------------------------------------------------------------------|
| RealServer Basic        | 25 users<br>free                                                                       |
| RealServer Plus:        | 60 concurrent users<br>\$ 1.995,95<br>RealAudio 8 (64 kbps)<br>RealVideo (VHS quality) |
| RealServer Professional | 100+ users<br>\$ 8339,00                                                               |
- Solaris SPARC, Linux x86, Windows x86
  - 256 .. 1024 MB RAM

Digitale Audioverarbeitung | WS 2000 | 18.205

## Liquid Audio

Digitale Audioverarbeitung | WS 2000 | 18.205

## Liquid Audio

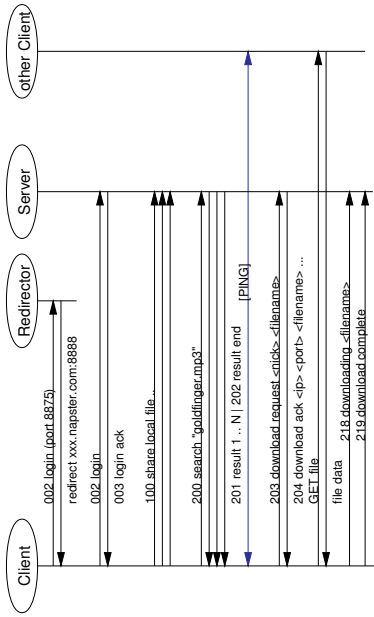
- viertes "big player" System: [www.liquidaudio.com](http://www.liquidaudio.com)
- Encoder: (Liquifier Pro)  
 Server: (Liquid Server)  
 Player: (Liquid Player)
- Datenformate MP3, AC-3, AAC, WMA
  - Verschlüsselung der Dateien (RSA)
  - digitales Wasserzeichen für Erkennung analoger Kopien
  - diverse Optionen, z.B. Start- und End- Gültigkeitsdatum
  - Player wird an die Hardware "gedongelt"
  - Dateien können nur auf einem Player abgespielt werden

Digitale Audioverarbeitung | WS 2000 | 18.205





## Napster



- Internet-Tauschbörse für MP3-Dateien
- Konzept natürlich auch allgemeiner einsetzbar

Digitale Audioverarbeitung | WS 2000 | 18.205

## Napster: Konzept

- einfaches Protokoll, ca. 30 Befehle
- eingeschränkt transparent für Firewalls

Napster-Server:

- Metadaten (z.B. user passwords)
- Index aller gerade verfügbaren "shared files"
- verarbeitet die Suchanfragen
- aber Downloads laufen direkt zwischen Clients
  - Benutzerprofile
  - detaillierte Statistiken
  - geringe Belastung der Server
- Servercluster für Lastausgleich und Skalierbarkeit (6M users)
- schwere Sicherheitslücke via Auto-Update des Clients

Digitale Audioverarbeitung | WS 2000 | 18.205

## Quicktime: Java API

```

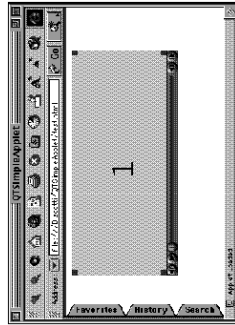
public class QTSimpleApplet extends Applet {
 private Drawable myQTContent;
 private Canvas myQTCanvas;

 public void init () {
 try {
 QTSession.open ();
 setLayout (new BorderLayout ());
 myQTCanvas = new QTCanvas (QTCanvas.kInitialSize, 0.5F, 0.5F);
 add (myQTCanvas, "Center");

 QTFile file = new QTFile (getCodeBase ().getFile (".getFile () +
 getParameter ("file")));
 myQTContent = QTFactory.makeDrawable (file);
 } catch (Exception qte) { ... }
 }

 public void start () {
 try {
 if (myQTCanvas != null)
 myQTCanvas.setClient (myQTContent, true);
 } catch (QTException e) { ... }
 }
}

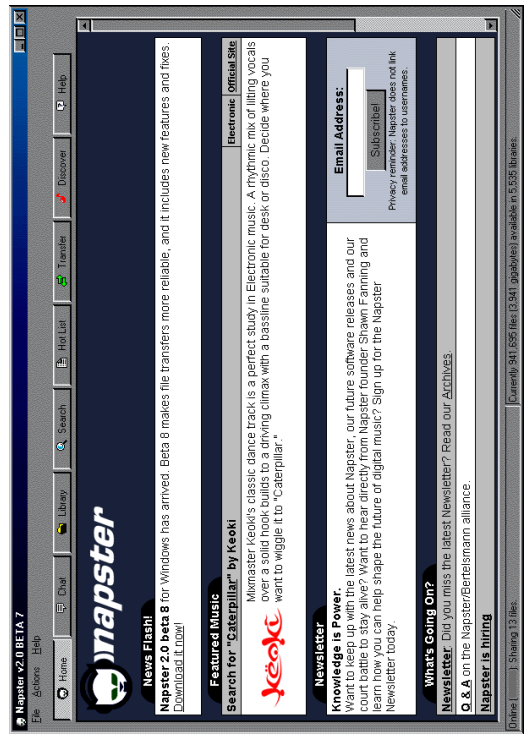
```



- "arguably simpler than C code"
- "C code differs for Mac and Windows"

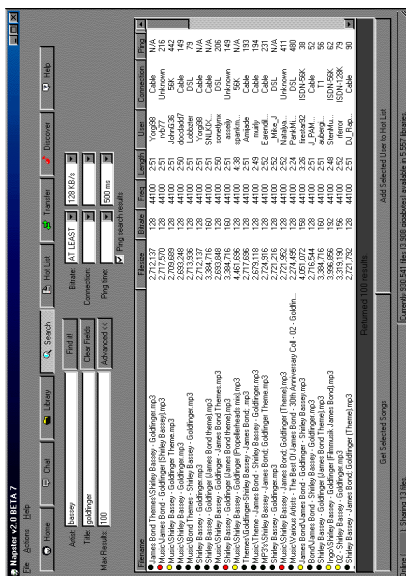
Digitale Audioverarbeitung | WS 2000 | 18.205

## Napster: Homepage



Digitale Audioverarbeitung | WS 2000 | 18.205

### Napster: Screenshot



- Suche: Anfrage und Ergebnisliste
- jeder Download wird vom Server protokolliert

### Napster: Login

- Protokoll setzt auf TCP/IP auf
  - Datenformat: <length> <type> <data>
    - o little endian byte order
- Beispiel login:
- length
  - <2>
  - <nickname>
  - <password>
  - <port>
  - <client info>
  - <link type>
  - [ <build number> ]
- "login" command type
- user nickname inside Napster
  - user password
  - data transfer port, 0=behind firewall
  - string with client version ID
  - 0=unknown, 1=14.4, 2=28.8, ... 10=T3
  - windows build version

### Napster: Client Search Request

- length
  - <200>
  - [ <artist name> ]
  - <max results>
  - [ <song name> ]
  - [ <linespeed op type> ]
  - [ <bitrate op type> ]
  - [ freq op type> ]
  - [ local only]
- "search" command  
filename contains "artist" | "song" substr  
0 .. 100 results returned  
same as artist name :-(  
0=unknown, 1=14.4 kbps, ...  
MP3 bitrate kbps  
MP3 samplerate  
search all / current Napster server
- 200 FILENAME CONTAINS "goldfinger" MAX\_RESULTS 25  
BITRATE "AT LEAST" "128" LINESPEED "AT LEAST" 10

### Napster: Search Response

- length
  - <201>
  - "filename"
  - <md5>
  - <size>
  - <bitrate>
  - <frequency>
  - <length>
  - <nickname>
  - <ip>
  - <link type>
  - [ <weight> ]
- "search response" command
- MD5 hash of file
  - file size in bytes
  - MP3 bitrate in kbps
  - MP3 samplerate
  - play time in seconds
  - the person sharing the file
  - the ip of the user with the file
  - 0= unknown, ...
  - positive= "better" match
- "goldfinger.mp3" 7d733...bcd2558199 128 44100 159  
"JamesBond" 134.100.13.183 4

## Streaming: Marktbedeutung Napster



### Über 6.7 Millionen Napster–User allein in den USA 106.10.2008 10:13:17

Die Musik–Tauschbörse **Napster**[1] entwickelt sich derzeit zur am schnellsten jennals über das Internet verbreiteten Software. Wie die Marktforscher von **MediaMatrix**[2] jetzt bekannt gaben, list die Zahl der User, die Napster von zu Hause aus benutzen, in den USA von 1,1 Millionen im Februar dieses Jahres auf über 6,7 Millionen im September gestiegen. Damit sind von den insgesamt knapp 80 Millionen US–Amerikanern, die von zu Hause aus im Internet surfen, 8,5 Prozent Napster–User. Und vom Arbeitsplatz aus frequentieren den Dienst zusätzlich über 1,1 Millionen Surfer. Welche Reichweite Napster weltweit hat, teilen die Marktforscher nicht mit.

Der Spalt für die Napster–User könnte aber bald vorbei sein. Derzeit kämpft die Tauschbörse von einem amerikanischen Berufungsgericht ums **Überleben**[3]. Die Vereinigung der amerikanischen Musikindustrie hatte gegen Napster wegen angeblicher Verstöße gegen Urheberrechtsbestimmungen geklagt. (axv/q/c't)

**URL dieses Artikels:**  
<http://www.heise.de/newsticker/data/axv-06.10.00-000/>

**Links in diesem Artikel:**  
 [1] <http://www.napster.com>  
 [2] <http://www.mmx.com>  
 [3] <http://www.heise.de/newsticker/data/k-03.10.00-001/>  
 [4] <mailto:axv@ct.heise.de>

- 6.7 Mio. (aktive) Benutzer
- Auswirkungen auf CD Verkäufe?!
- Tauschbörse / Raubkopien ?!

## Streaming: Marktbedeutung ?!

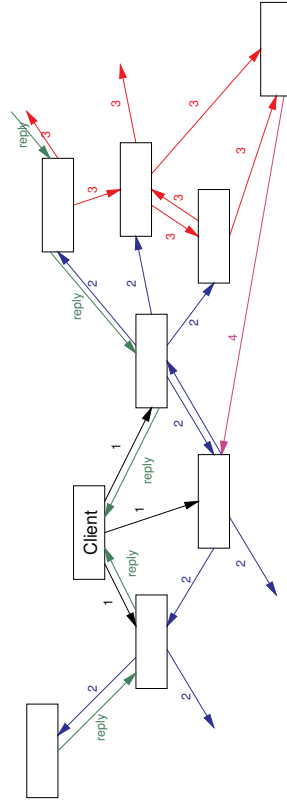


[www.modernhumorist.com]

## Gnutella

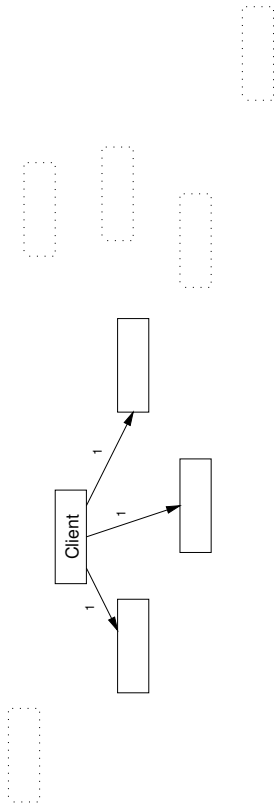
- Protokoll zum Datenaustausch zwischen vernetzten Rechnern
- definiert Suchanfragen und eigentlichen Transfer (via HTTP)
- völlig verteilte Architektur, kein zentraler Server (!)
- jeder Client dient gleichzeitig als Server und Suchmaschine  
 =: "servant" (servent?)
- entwickelt von Nullsoft (Nullsoft, WinAMP)
- nach Kauf von AOL sofort eingestellt
- aber bereits "überall" im Internet erhältlich, gnutella.wega.com
- reverse-engineering des Protokolls, Clients für Win/Mac/X11/Java/...
- "the source code has not been released yet [...], due to AOL locking it in a closet underneath a large pile of Time Warner's dirty laundry"  
 Gnutella 0.56 FAQ

## Gnutella: Suchstrategie



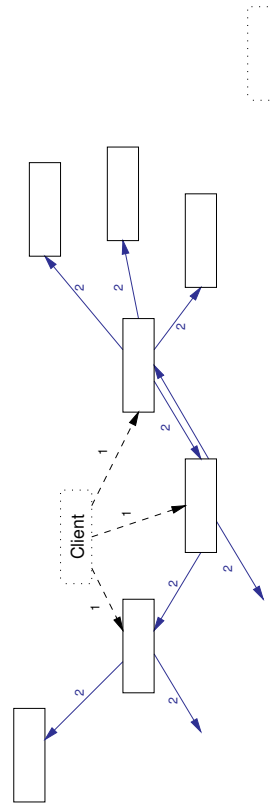
- Client befragt seine unmittelbaren Nachbarn, z.B. mit TTL=25
- diese wiederum ihre unmittelbaren Nachbarn, --TTL
- => Paketflut (exponentielles Wachstum!?)
- => Kennzeichnung jeder Anfrage mit GUID
- => jeder Servant bearbeitet eine Anfrage (höchstens) einmal
- Anfragen durchsuchen gesamtes Gnutella-Net

### Gnutella: Suchstrategie (1)



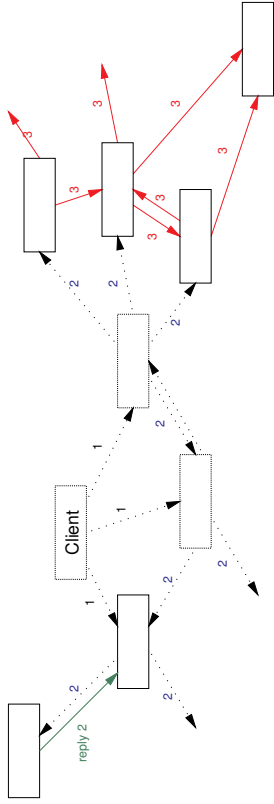
- Client befragt seine unmittelbaren Nachbarn, TTL=25
- Anzahl der Nachbarn abhängig von verfügbarer Bandbreite
  - schnelle Hosts erlauben viele Verbindungen
  - Modem/ISDN-Hosts mit wenigen Verbindungen

### Gnutella: Suchstrategie (2)



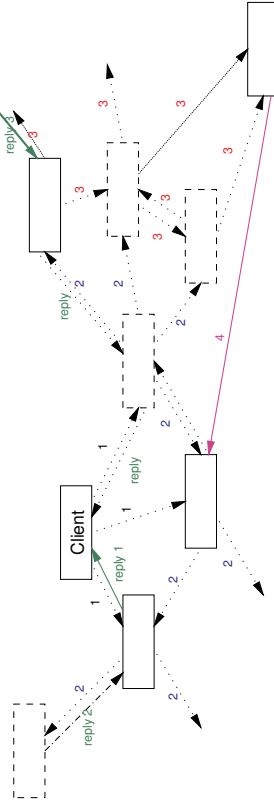
- Client befragt seine unmittelbaren Nachbarn, TTL=25
  - diese wiederum ihre unmittelbaren Nachbarn
  - ausgenommen den fragenden Rechner selbst
- => Paketflut (exponentielles Wachstum, zirkuläre Anfragen)
- => Kennzeichnung jeder Anfrage mit GUID
- => jeder Servant bearbeitet eine Anfrage (höchstens) einmal

### Gnutella: Suchstrategie (3)



- Anfrage erreicht alle sichtbaren Nachbarn
- in wenigen Zyklen
- Suchantwort läuft umgekehrten Weg zurück
- möglich wegen GUIDs und TTL-Sortierung
- Zurückverfolgen der Suche stark erschwert

### Gnutella: Suchstrategie (4)



- Anfragen durchsuchen gesamtes Gnutella-Net
- Antwort kann aber recht lange dauern (mehrere Minuten)
- aktuelle Clients überlasten das Netz durch zu viele PINGS

### Gnutella: Protokoll

- textbasiertes Protokoll auf HTTP
- fünf "Pakettypen" definiert:

| Paket | Funktion                         |
|-------|----------------------------------|
| 0x00  | Ping                             |
| 0x01  | zur Suche nach Gnutella-Servants |
| 0x02  | Antwort eines Servants           |
| 0x80  | Suchanfrage nach Daten           |
| 0x81  | Suchergebnis mit IP/Dateiname    |
| 0x40  | Datenübertragung via Firewalls   |

| GUID | type | TTL | hops | length | payload ... |
|------|------|-----|------|--------|-------------|
| 0    | 15   | 16  | 17   | 18     | 19          |
|      | 20   | 21  | 22   | 23     | ...         |

- diverse Erweiterungen / Verbesserungen vorgeschlagen
- eigentlicher Download erfolgt direkt (peer-to-peer)

### Gnutella: Pong, Query, QueryHit

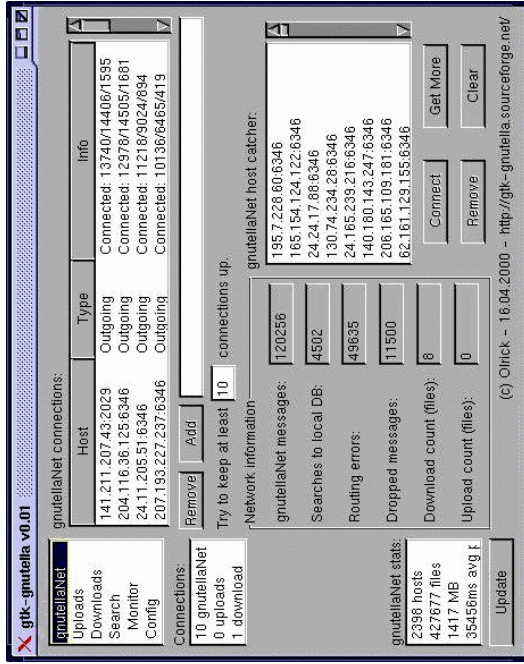
| Port | Host IP | # shared files | # shared kBytes |
|------|---------|----------------|-----------------|
| 0    | 1       | 2              | 3               |
|      | 4       | 5              | 6               |
|      | 7       | 8              | 9               |
|      | 10      | 11             | 12              |
|      | 13      | 14             | 15              |

| min. speed | search criteria |
|------------|-----------------|
| 0          | 1               |
|            | 2               |

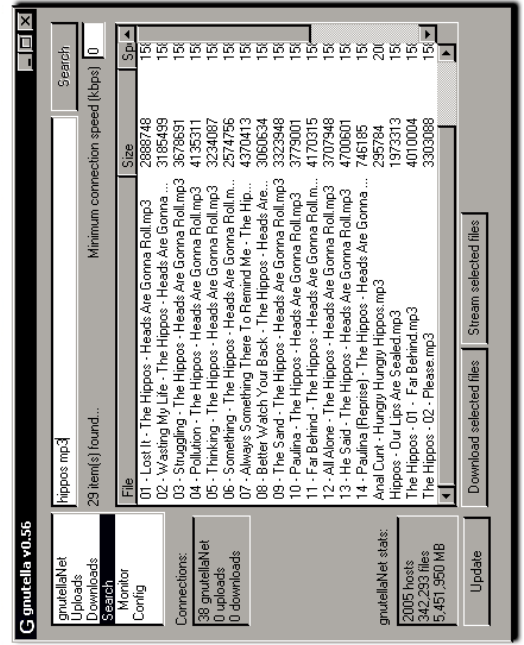
| # hits | Port | Host IP | Speed | result set | servant ID |
|--------|------|---------|-------|------------|------------|
| 0      | 1    | 2       | 3     | 4          | 5          |
|        | 6    | 7       | 8     | 9          | 10         |
|        | 11   | 12      | 13    | 14         | 15         |
|        | 16   | 17      | 18    | 19         | ...        |
|        | 20   | 21      | 22    | 23         | n          |

- Host (=Servant) IPs in den PONG und QueryHit Antworten
- Zurückverfolgung also im Prinzip doch möglich

### Gnutella: Screenshot (GTK-Version)

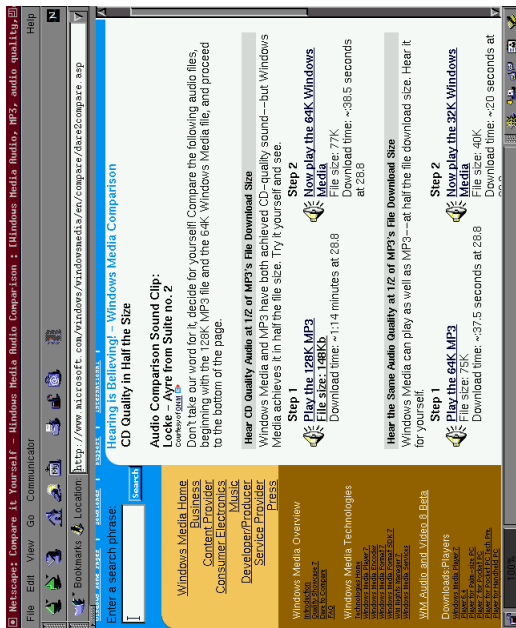


### Gnutella: Screenshot Suchanfrage



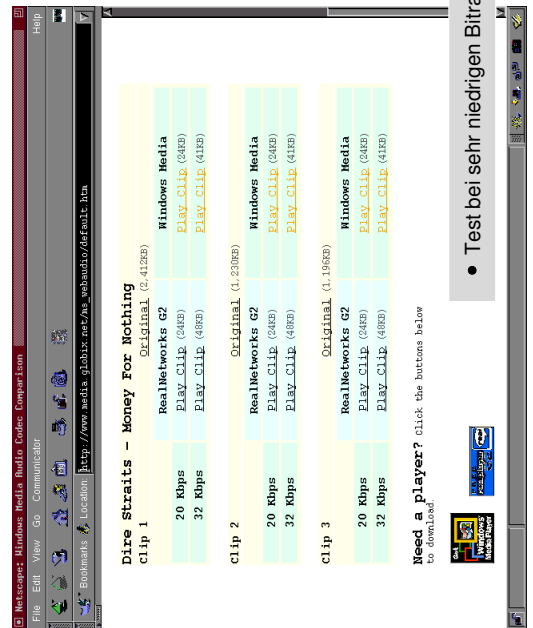


## Vergleich: Microsoft WMA vs. MP3



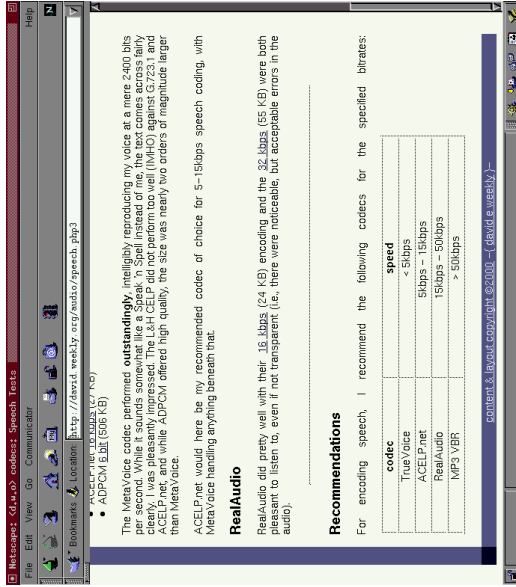
Digitale Audioverarbeitung | WS 2000 | 18.205

## Vergleich: Microsoft WMA vs. G2



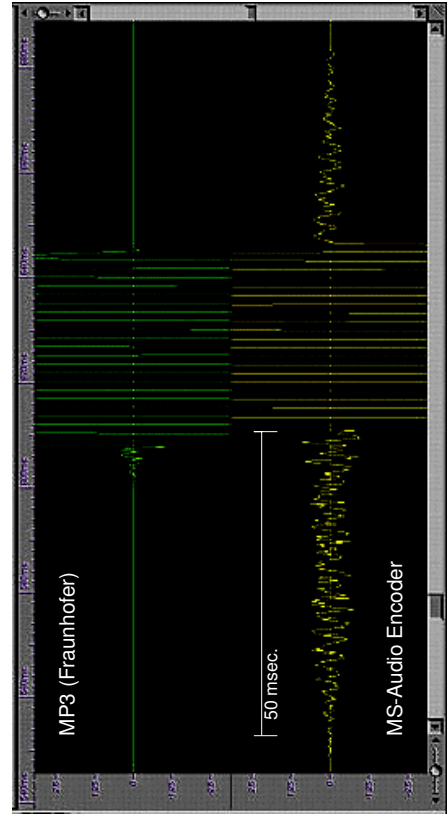
Digitale Audioverarbeitung | WS 2000 | 18.205

## Vergleich: Speech Codecs



Digitale Audioverarbeitung | WS 2000 | 18.205

## Pre-Echos: MP3 vs. MS Audio



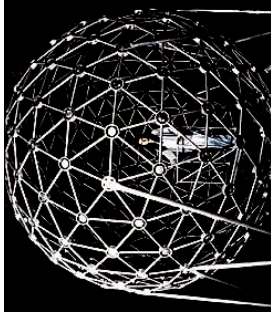
- krasse Beispiel für "pre-echos" (c't 99/10 p.52)
- aktuelle Encoder arbeiten (hoffentlich) besser

Digitale Audioverarbeitung | WS 2000 | 18.205

- Test bei sehr niedrigen Bitraten

## 3D-Audio: Agenda

- akustische Raumwahrnehmung:
  - Lautstärke- und Zeitdifferenzen
  - Klangdifferenzen, HRTF
  - Raumumgebung, Reflexionen, Hall
- Mehrkanalverfahren
- Stereo bis Dolby AC-3 und DTS
- Microsoft DirectSound
- Creative EAX
- Sensaura
- Miles Sound System Demo



277 speakers... [Physics Today Nov/99]

## 3D-Audio: Literatur

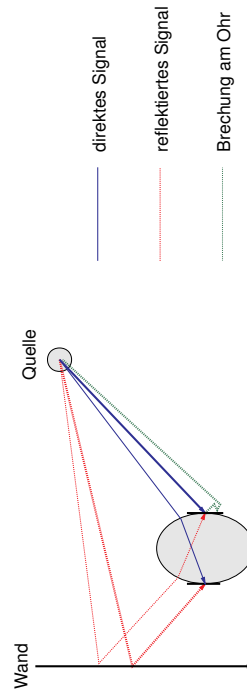
- Blauert, Spatial Hearing  
Raumwahrnehmung. [www.aip.org/pt/nov99/ocsound.html](http://www.aip.org/pt/nov99/ocsound.html)
- Dolby Labs Whitepaper, [www.dolby.com/tech/dts/](http://www.dolby.com/tech/dts/)
- B. Barga and P. Donnelly, Inside DirectX, MS Press, ISBN 3-86063-436-4
- Microsoft DirectSound Docs (im DirectX7 / DirectX8 SDK)
- MIDI Manufacturers Association, Interactive 3D Audio Rendering Guidelines
- Creative EAX 1.0 und EAX 2.0, [developer.soundblaster.com](http://developer.soundblaster.com)
- Miles Sound System, [www.smacker.com/mss3d.htm](http://www.smacker.com/mss3d.htm)
- Sensaura 3D-Audio Dokumentation, [www.sensaura.com](http://www.sensaura.com)

## Akustische Raumwahrnehmung

Raumwahrnehmung beruht auf:

- Schallintensität am linken / rechten Ohr
  - Zeit- und Phasendifferenzen
  - Außenohr- und Kopfübertragungsfunktion
  - "interaural level/intensity/time difference"
  - Ankunftszeit der Schalls ("precedence effect")
  - Reflexionen / Dämpfung des Schalls
  - erwartete Position der Quelle (z.B. Hundegebell: am Boden)
- => außerordentliche komplexe Verarbeitung im Gehirn
- => Nachbildung / Rekonstruktion mit Signalverarbeitung
- => Approximation durch Vielzahl von Lautsprechern

## Akustische Raumwahrnehmung

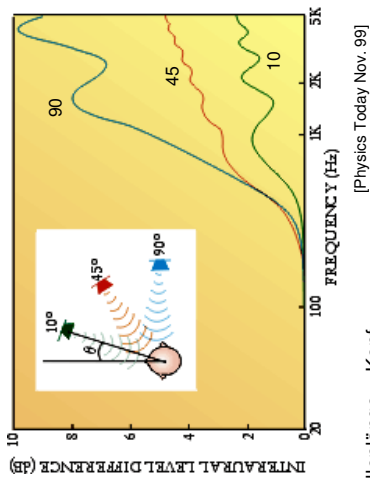


- direktes Signal
- reflektierte Signale
- jeweils mit Beugung um den Kopf herum
- jeweils mit Brechung an Ohr, Schultern, ...
- zusätzlich "obstruction" und "occlusion" für verdeckte Quellen ...



### ILD: Lautstärkedifferenz

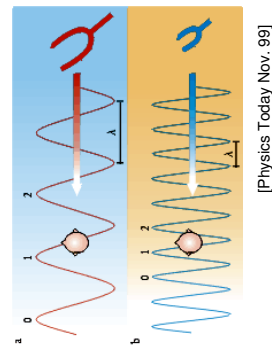
interaural level difference:



- Welle kaum gestört, wenn Wellenlänge > Kopf
- daher kaum Effekt bei niedrigen Frequenzen
- Bässe schlecht ortbar

### ITD: Zeit- und Phasendifferenz

interaural time difference:



$$dt = \frac{3a}{c} \sin \theta$$

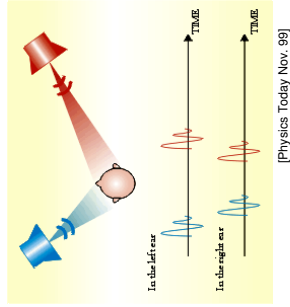
a = Kopfradius ~ 8.75 cm

c = Schallgeschw ~ 340 m/s

- 500 Hz Sinuston von vorne ortbar mit ca. 1 Grad Auflösung => ca. 13 ms Zeitauflösung
- mehrdeutig, wenn n \* Wellenlänge ~ = Kopfdurchmesser
- Ohr unempfindlich für Phaseninformation oberhalb ca. 1 KHz
- vgl. Hilbertfilter-Experiment

### Ankunftszeit

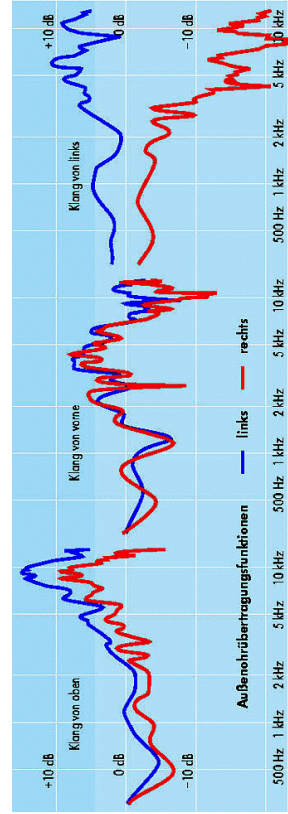
Experiment zum "precedence effect":



- Wiedergabe eines Impulssignals über zwei Lautsprecher (Monosignal auf Stereoanlage)
- linkes Signal am linken Ohr früher
- rechtes Signal am rechten Ohr früher
- aber insgesamt linkes Signal früher
- sehr starker Einfluß auf die Interpretation des Signals

=> Wahrnehmung als ein einzelnes Signal von links

### HRTF



"head related transfer function": Außenohr/Kopf-Übertragungsfunktion

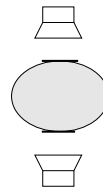
- Einfluß von Kopf, Schultern, Außenohr auf Schallsignale
- sehr komplexe Winkel- und Frequenzabhängigkeit
- Verbesserung der Ortung durch (kleine) Kopfbewegungen

**HRTF: Messung**



- Kleinstmikrophone
- Kunstkopf
- im schalltoten Raum
- für alle Raumwinkel
- KEMAR-Experiment und Daten: [sound.media.mit.edu/KEMAR.html](http://sound.media.mit.edu/KEMAR.html)
- oft Verwendung "gemittelter" HTRF (z.B. DirectSound)
- mit "typischen" Filterverläufen

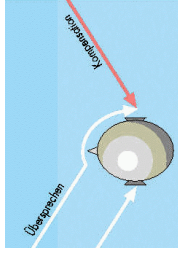
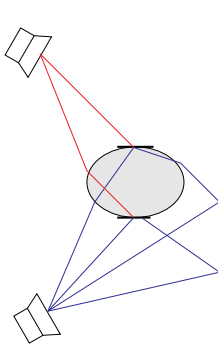
**HRTF: Kopfhörer**



im Prinzip perfekter Raumklang erreichbar:

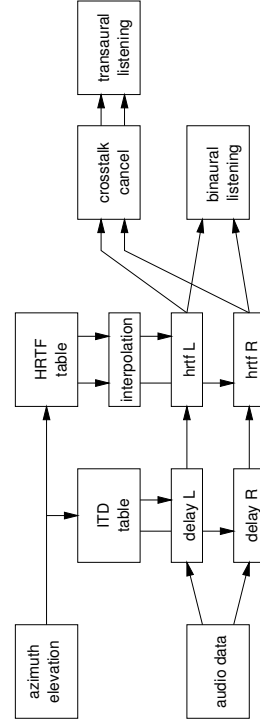
- direkte Schallübertragung zum Ohr, (rel.) Position des Hörers bekannt
- fast kein Übersprechen, keine externen Reflexionen
- durch HRTF perfekt kompensierbar
- aber generische HRTF reicht dazu nicht aus
- ideal wäre die Messung der einzelnen HRTF
- Tools / Geräte dazu ?!

**HRTF: Lautsprecher**



- Übersprechen zwischen den Lautsprechern
- Störung durch Raumreflexionen
- Head-Tracking erforderlich (Position und Richtung)
- wesentlich aufwendiger als über Kopfhörer
- aber im Ansatz möglich, z.B. [www.sensaura.com](http://www.sensaura.com)

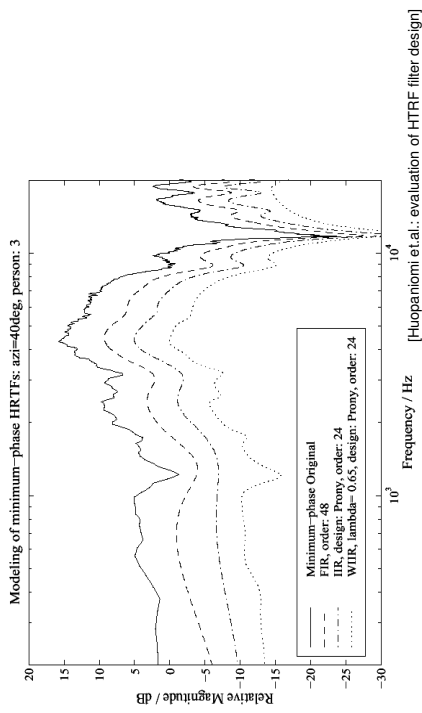
**HRTF: Umsetzung**



Signalverarbeitung kann alle Teilinformationen liefern:

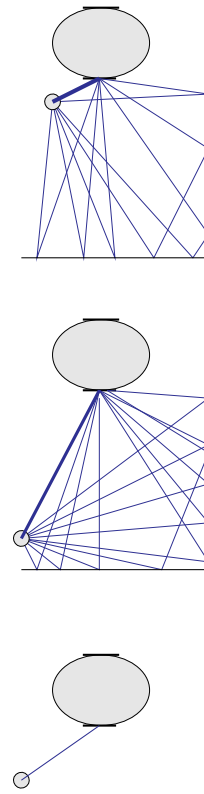
- Lautstärke, Verzögerung, HRTF
- evtl. zusätzliche Unterdrückung des Übersprechens
- Details und Diskussion: Huopaniemi et.al.: "evaluation of HTRF filter design" [CCRMA]

### HRTF: Filter-Design



- entsprechende Filterkurven für jeden Winkel notwendig (!)

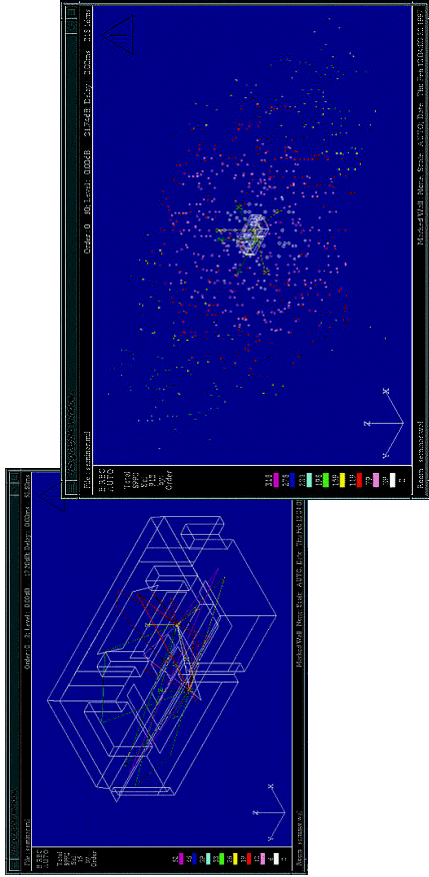
### Reflexionen



#### Reflexionen / Hall:

- abhängig von Raumgröße, -geometrie, Wandmaterial, ...
- sehr wichtig für "virtual reality": Zimmer, Bad, Tunnel, Kathedrale, ...
- indirekter Schall unabhängig von Entfernung Quelle/Hörer
- Entfernungsmessung über Anteil direkt/reflektierter Schall
- Modellierung über Spiegelquellen (=raytracing)
- oder pauschal über Hallparameter

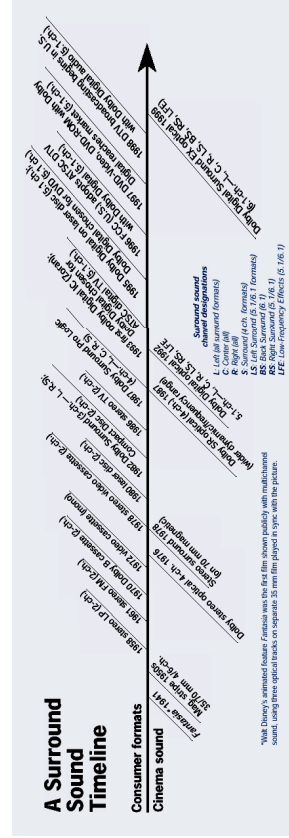
### Spiegelquellen ...



- Modellierung von Konzertsälen etc.
- zu rechenaufwendig für Echtzeit

[et 04:57:160]

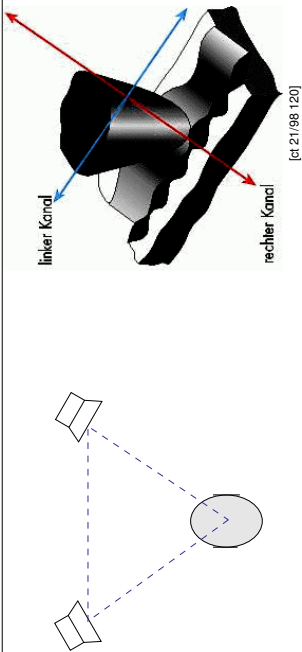
### Mehrkanalton: Geschichte



[Dolby Labs: Surround Sound: Past, Present, and Future]

- erste HRTF-Messungen ca. 1940 (?)
- DirectSound3D seit 1997 (DirectX5)
- Erweiterungen wie EAX und A3D seit 1998

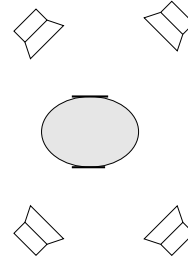
### Mehrkanalton: Stereo und LP



Stereo:

- zwei unabhängige Kanäle für Aufzeichnung und Wiedergabe
- Tonaufnahmen seit ca. 1950, LPs und UKW ca 1960
- gute Wiedergabe typischer Konzertsituation
- benötigt nur zwei Lautsprecher
- optimale Hörposition oft sehr klein: "sweet spot"

### Mehrkanal: Quadrophonie



- Wiedergabe von vier Kanälen
- mit LP nicht möglich, daher kaum erhältlich
- "surround"-Signal enthält nur die "live" Störsignale ...
  - für Musik / Konzerte nicht allzu nützlich
  - anders als für Wiedergabe von Filmen / 3D-Spielen

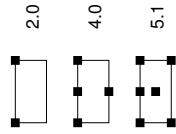
### Kino: Überblick

Film als "Erlebnis" erfordert "Eintauchen":

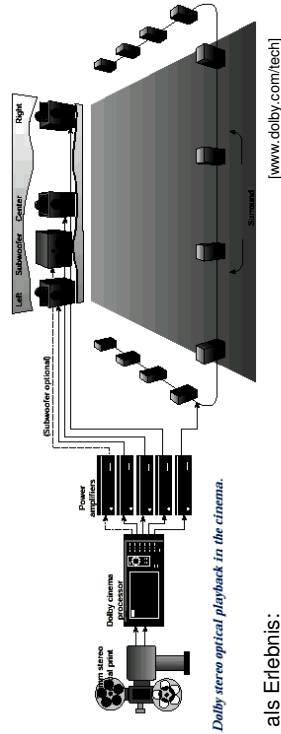
- möglichst gute Übereinstimmung von Bild und Ton
- Surround-Signale für Umgebungsgeräusche
- problematische Hörpositionen für die meisten Zuschauer
- Übersteigerung statt virtueller Realität:

- LFE: low frequency effects, Infraschalleffekte
- Beeinflussung durch emotional belegte Klänge
- verhältnismäßig hohe Lautstärke

- => Trendsetter für Mehrkanalton
- => diverse erfolgreiche Verfahren, seit ca. 1980
- X:Y Notation: X vollwertige Kanäle, Y LFE-Kanäle
- typisch ist 5.: L/C/R, SL/SR, LFE



### Kino: Mehrkanalton

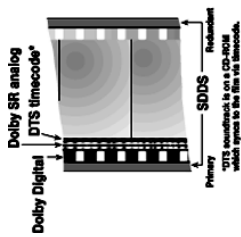


Film als Erlebnis:

- Übereinstimmung von Bild und Ton erforderlich
- Stereo unzureichend: "sweet spot" zu klein
- zusätzlicher Center-Kanal für Stimmen
- verbesserte "Immersion" durch Surround-Signale aber wohin mit den Tonspuren?
- Lichtton optisch, direkt neben den Bildern
- zusätzliche Magnetspuren auf dem Film (teuer)

## Dolby Surround

- erstes Surroundformat für 35mm Film
- zwei Lichtspuren neben dem Bild
- statt bisheriger Monospur
- Dolby-A Rauschunterdrückung

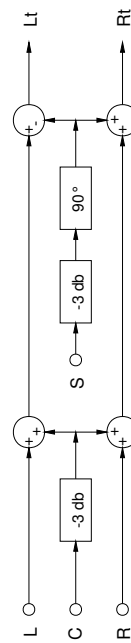


**Current soundtrack formats**  
[www.dolby.com/tech]

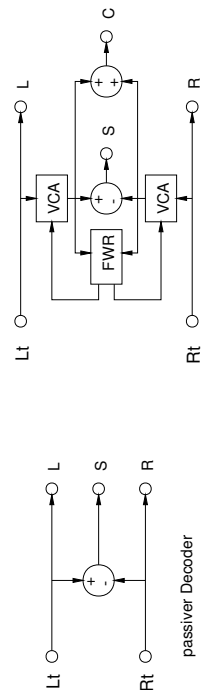
"Matrix"-Kodierung

- vier Eingangssignale: links, rechts, center, surround
- auf Stereo-Ausgangssignal
- Audioqualität (technisch) nicht optimal
- aber besseres Kinoerlebnis
- Stereo- und Mono-kompatibel

## Dolby Surround: Prinzip

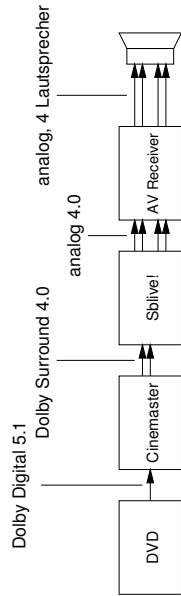


- Center-Kanal wird auf L/R zugemischt (Stereo/Mono-kompatibel)
- Surround-Kanal phasengedreht zugemischt (off center)
- Surround-Kanal zusätzlich bandbegrenzt (100 .. 7 KHz)

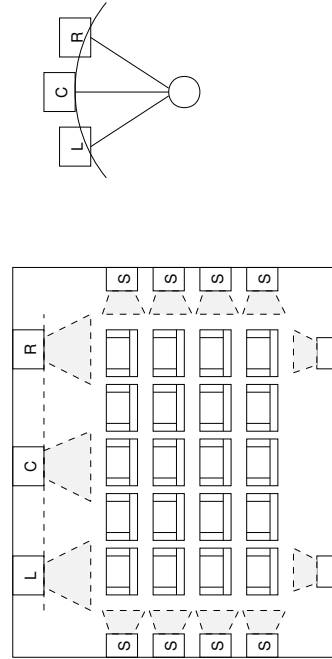


## Dolby Surround: Demo

- Audioformat auf vielen älteren DVDs
- z.B. englisch in Dolby Surround, andere Sprachen mono ...
- auch als Notlösung für PC Software-DVD Player:
- Signallauf: (suboptimal)

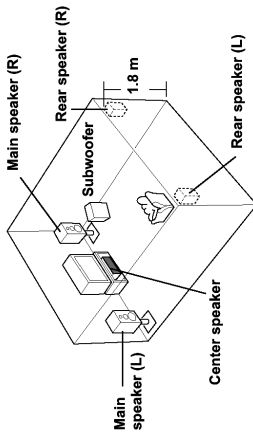


## Dolby: Speaker Placement



- Frontlautsprecher ideal in gleicher Höhe
- Centerlautsprecher nach hinten versetzt (gleicher Hörabstand)
- Surroundlautsprecher links und rechts, diffuse Abstrahlung
- Subwoofer nach Bedarf, oder gute Frontlautsprecher

## Dolby: Speaker Placement

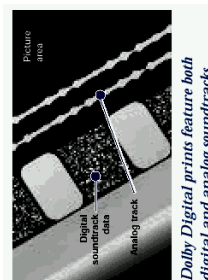


Aufstellung zuhause:

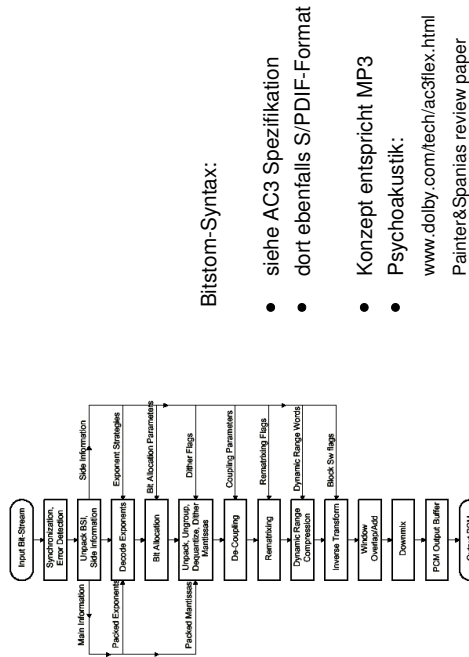
- Hauptlautsprecher "normal"
- Center direkt über/unter dem Bild
- Surroundlautsprecher seitlich oder an der Rückwand
- Subwoofer seitlich vorne
- auch für 3D-Spiele gut geeignet
- Vortführung mit Creative Sblivel: Experience Demo

## Dolby Digital (AC-3)

- digitale Aufzeichnung auf 35 mm Film
- zwischen den Transportlöchern
- zusätzlich Dolby Surround Spuren
- psychoakustische Aufzeichnung
- ähnliches Konzept wie MP3
- fünf Audiokanäle plus LFE-Kanal: 5.1
- Bitrate 320 .. 384 kbps
- begrenzt durch opt. Auflösung des 35 mm Films
- deutlich bessere Qualität als Dolby Surround
- zwei separate Surroundkanäle, ideale Kanaltrennung
- dominierendes Audioformat für aktuelle DVDs
- Erweiterung auf 6.1 als Dolby Digital ES (1999)



## Dolby Digital: Bitstrom und Decoder



Bitstrom-Syntax:

- siehe AC3 Spezifikation
- dort ebenfalls S/PDIF-Format
- Konzept entspricht MP3
- Psychoakustik: [www.dolby.com/tech/ac3flex.html](http://www.dolby.com/tech/ac3flex.html)  
Painter&Spanias review paper

## Dolby Digital: Demo



Vergleich: Stereo / Dolby Surround / Dolby Digital:

- Man-in-Black, Szene 9 (Surround-Effekte)
- Arthaus Musik-Sampler (Klassik, Konzert und Oper)
- Lola rennt (Techno)

## DTS

Mehrkanalsystem von "Digital Theatre Systems"

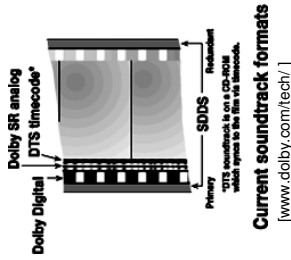
- geringere Kompression als Dolby AC3
- im Prinzip bessere Klangqualität
- direkter Vergleich derzeit kaum möglich

Kino:

- Audiodaten auf CD-ROM
- Synchronisation über Timecode auf dem Film

DVD:

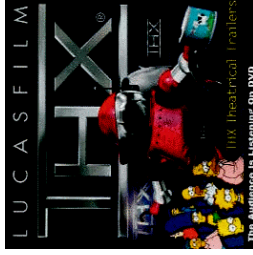
- DTS-DVDs mit 768 kbps (selten 1.5 Mbps)
- zusätzliche Tonspur, z.B. Stereo oder Dolby Stereo 2.0
- wird von allen DVD-Playern unterstützt



## THX:

Lucasfilm THX, [www.thx.com](http://www.thx.com):

- kein eigenes Audioformat
- sondern Richtlinien für Klang (-abstimmung)
- THX-Logo nur bei Erfüllen aller Anforderungen
- Liste getesteter Geräte (decoder, amps, speaker, ...)
- getrennt für Profi (=Kino) und Home-Markt
- THX-Klang für Musikkwiedergabe weniger geeignet
- THX-Demo DVD leider nur für "licencees" verfügbar



## MPEG-2: AAC

diverse MPEG Audioformate:

- MPEG-1 layer 1-2-3: Stereosignale
- MPEG-2 BC: wie MP3, aber Mehrkanalfähig
- MPEG-2 AAC: "advanced audio coding"
  - verbesserte Kodierung
  - (niedrigere Bitraten)
  - Mehrkanal bis 24.7
  - für DVDs bis 7.1 vorgesehen (bisher selten verwendet)

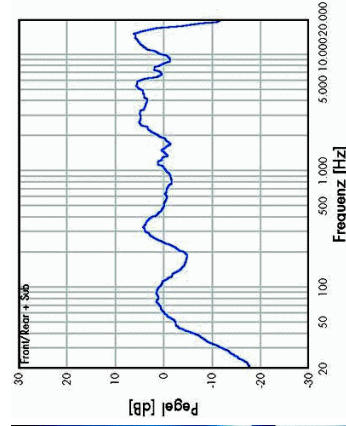
Details zu Algorithmen und Bitstreams: siehe perceptual coding

- MPEG-4:
  - structural audio: portable Synthesialgorithmen

## Lautsprecher: Beispiel Crossfire

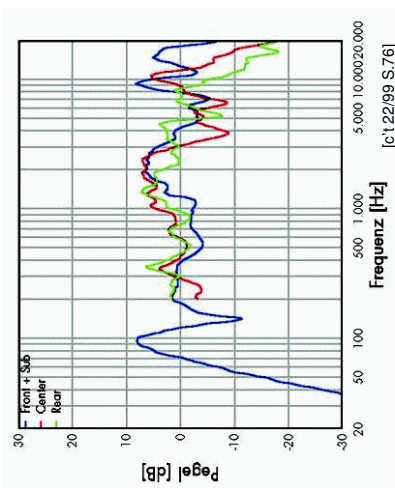


- Videologic "Crossfire" System
- Subwoofer mit 4 Satelliten



[c: 26.99 S.70]

### Lautsprecher: Beispiel Digi Theatre



- Digi Theatre 5.1

### Mehrkanalton: Raum F334

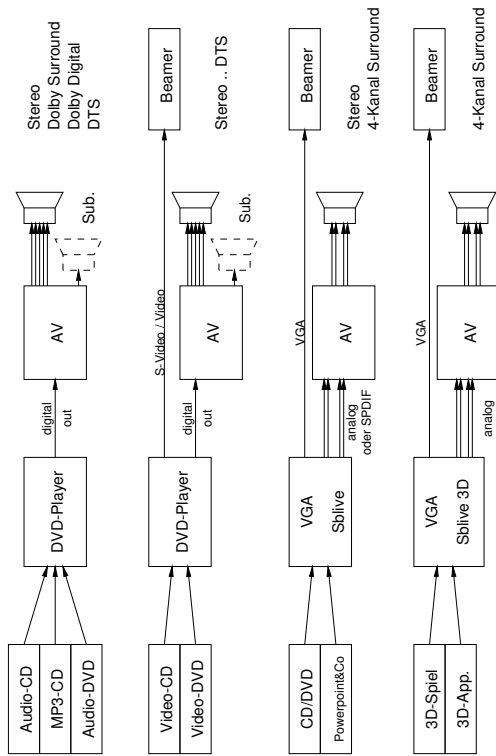
#### Geräte:

- Beamer Epson 7350
- AV-Receiver Yamaha RX596
- DVD-Player Aiwa XD DV 370

#### Lautsprecher:

- Front Canton CT-220
- Center Canton AV-950
- Surround Canton CT-220
- Subwoofer Canton AS-25
- normale statt "Dipol"-Surroundlautsprecher
- Aufstellung / Ausrichtung noch nicht optimiert

### F334: Multimedia-Optionen



### AV-Receiver: Yamaha RX-596

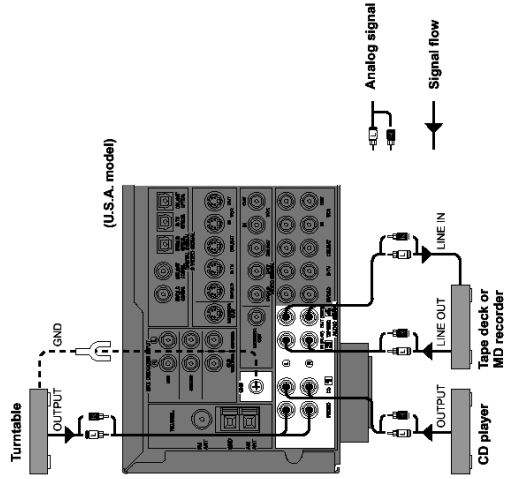


#### "typischer" Home AV-Receiver:

- Verstärker mit 5 x 100 W
- Subwoofer-Ausgang
- Stereo, Dolby Surround, Dolby Digital, DTS
- externer Decoder (5.1) möglich
- Digitaleffekte ("Movie spectacle", "sound fields")
- Sharp DSP
- Video-Umschalter
- insgesamt 5 Video- und 3 zusätzliche Audioeingänge
- ca. DM 1000.00

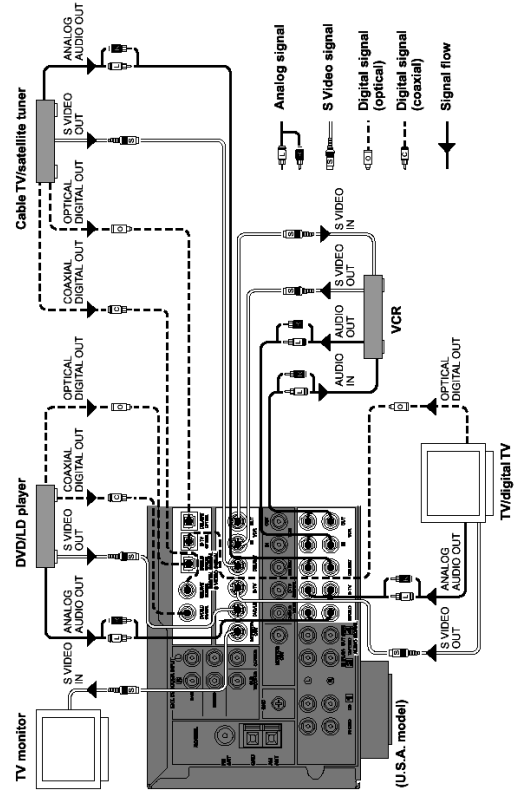


RX-596: Audio



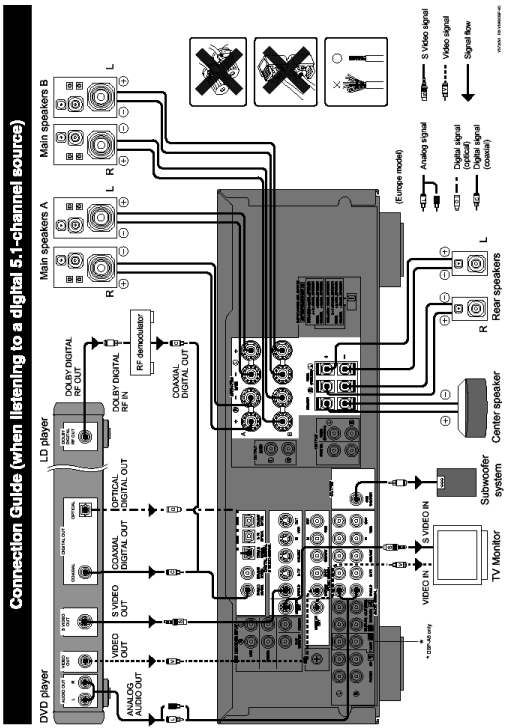
Digitale Audioverarbeitung | WS 2000 | 18.205

RX-596: Video



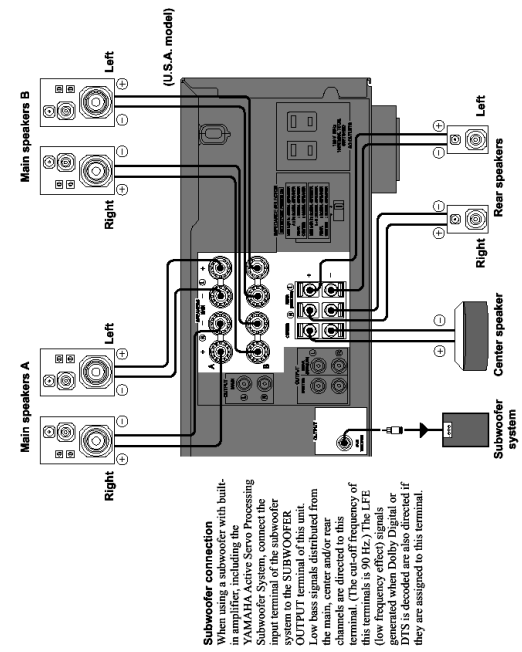
Digitale Audioverarbeitung | WS 2000 | 18.205

RX-596: Anschlüsse



Digitale Audioverarbeitung | WS 2000 | 18.205

RX-596: Lautsprecher



Digitale Audioverarbeitung | WS 2000 | 18.205

## RX-596: Composite Video

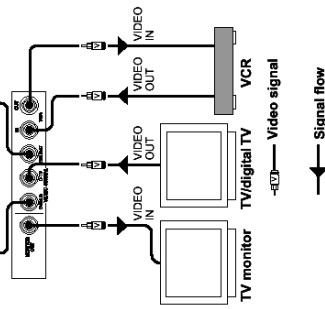
### VIDEO terminals (composite)

DVD/LD player Cable TV/satellite tuner

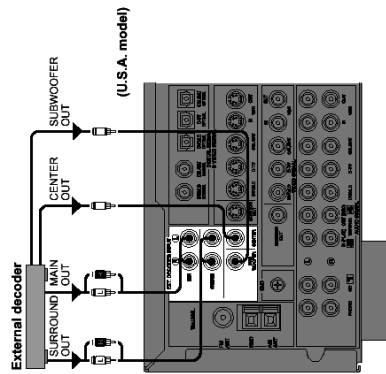
If your video components do not have "S" video terminals, they can be connected to this unit's VIDEO terminals. Be sure to connect the input (IN) and output (OUT) properly.

**Note**

- If video signals are input from both the S VIDEO input and composite input terminals, the signals will be directed to their respective output terminals.



## RX-596: External Decoder



This unit has additional 6-channel audio signal input terminals for connecting an external decoder to this unit. Connect the 6-channel audio signal output terminals of the decoder to the EXTERNAL DECODER INPUT terminals of this unit.

**Notes**

- When a source connected to these terminals is selected, the digital sound field processor cannot be used.
- The settings of "CENTER SP", "REAR SP", "MAIN SP" and "BASS OUT" on the SET MENU have no effect on a source connected to these terminals. The setting of "MAIN LVL" is effective. (Refer to pages 36 and 37 for details.)
- Adjustment of the output level of the center speaker, rear speakers and subwoofer is effective when a source connected to these terminals is selected as the input source. (Refer to page 39 for details.)

## RX-596: Soundfields

### CINEMA DSP: Dolby Surround + DSP/Dolby Digital + DSP/DTS + DSP

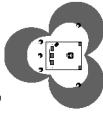
#### ■ Dolby Pro Logic + 2 digital sound fields

Digital sound fields are created in both the presence and rear surround zones of the Dolby Pro Logic-decoded sound field. They create a wide acoustic environment and emphasize the surround effect in the room, letting you feel as much presence as if you were watching a movie in a popular Dolby Stereo theater.



#### ■ Dolby Digital or DTS + 3 digital sound fields

Digital sound fields are created in the presence zone and independently on the left and right surround zones of the Dolby Digital-decoded or DTS-decoded sound field. They create a wide acoustic environment and strong surround effect in the room without losing high channel separation. With the wide dynamic range of Dolby Digital or DTS sound, this sound field combination lets you feel as if you were watching a movie in the newest Dolby Digital theater or DTS-installed theater. This is the most ideal home theater sound at the present time.



"Klangfeldprogramme":

- künstlich erzeugte Raumakustik, z.B. Nachhall
- Presets für "enhanced Dolby", "movie theater", "disco", ...
- wie Stereo-Basisverbreiterung / Super-Bass / etc.

## Direct Sound: Übersicht

DirectSound: [www.microsoft.com/directx/](http://www.microsoft.com/directx/)

- Microsoft API zur Audioverarbeitung
- als Ersatz für direkte SB16 Registerprogrammierung
- Abspielen und Mischen von beliebig vielen wav Quellen
- automatische Verwaltung der benötigten Puffer
- erkennt und benutzt vorhandene Hardware-Funktionen
- Hardware-Emulation in Software, wenn notwendig
- automatische Samplerate-Konvertierung
- 3D-Funktionen inklusive HRTF
- (bisher nur) ein (Stereo-) Ausgang
- entwickelt hauptsächlich für Spiele
- für "Profi"-Anwendungen (Tonstudios) weniger geeignet
- MIDI-Funktionen mittlerweile (DX7) über DirectMusic



## DirectSound: IDirectSound3DBuffer

```
// IDirectSound3DBuffer
//
// DEFINE_GUID (IID_IDirectSound3DBuffer, 0x279AFA86, 0x4981, 0x11CE,
// 0xA5, 0x21, 0x00, 0xAF, 0x0B, 0xE5, 0x60);
//
DECLARE_INTERFACE_(IDirectSound3DBuffer, IUnknown)
{
 // IUnknown methods
 STDMETHOD(QueryInterface)
 (THIS) PURE;
 STDMETHOD_(ULONG, AddRef)
 (THIS) PURE;
 STDMETHOD_(ULONG, Release)
 (THIS) PURE;

 // IDirectSound3DBuffer methods
 STDMETHOD(GetAllParameters)
 (THIS_ LPDS3DBUFFER) PURE;
 STDMETHOD(GetConeAngles)
 (THIS_ LPDWORD, LPDWORD) PURE;
 STDMETHOD(GetConeOrientation)
 (THIS_ LPDWORD, LPDWORD) PURE;
 STDMETHOD(GetConeOutSideVolume)
 (THIS_ LPDWORD, LPDWORD) PURE;
 STDMETHOD(GetMaxDistance)
 (THIS_ LPD3DVALUE) PURE;
 STDMETHOD(GetMinDistance)
 (THIS_ LPD3DVALUE) PURE;
 STDMETHOD(GetMode)
 (THIS_ LPDWORD) PURE;
 STDMETHOD(GetPosition)
 (THIS_ LPD3DVECTOR) PURE;
 STDMETHOD(GetVelocity)
 (THIS_ LPD3DVECTOR) PURE;
 STDMETHOD(SetAllParameters)
 (THIS_ LPD3DBUFFER, DWORD) PURE;
 STDMETHOD(SetConeAngles)
 (THIS_ DWORD, DWORD, DWORD) PURE;
 STDMETHOD(SetConeOrientation)
 (THIS_ D3DVALUE, D3DVALUE, D3DVALUE, DWORD) PURE;
 STDMETHOD(SetConeOutSideVolume)
 (THIS_ LONG, DWORD) PURE;
 STDMETHOD(SetMaxDistance)
 (THIS_ D3DVALUE, DWORD) PURE;
 STDMETHOD(SetMinDistance)
 (THIS_ D3DVALUE, DWORD) PURE;
 STDMETHOD(SetMode)
 (THIS_ D3DVALUE, D3DVALUE, D3DVALUE, DWORD) ;
 STDMETHOD(SetPosition)
 (THIS_ D3DVALUE, D3DVALUE, D3DVALUE, DWORD) ;
 STDMETHOD(SetVelocity)
 (THIS_ D3DVALUE, D3DVALUE, D3DVALUE, DWORD) ;
};
```

Digitale Audioverarbeitung | WS 2000 | 18.205

## DirectSound: Codebeispiel

```
VOID SetParameters (FLOAT fDopplerFactor, FLOAT fRolloffFactor,
 FLOAT fMinDistance, FLOAT fMaxDistance)
{
 // Every change to 3-D sound buffer and listener settings causes
 // DirectSound to remix, at the expense of CPU cycles.
 // To minimize the performance impact of changing 3-D settings,
 // use the DS3D_DEFERRED flag in the dwApply parameter of any of
 // the IDirectSound3DListener or IDirectSound3DBuffer methods that
 // change 3-D settings.
 // Then call the IDirectSound3DListener::CommitDeferredSettings
 // method to execute all of the deferred commands at once.
 DWORD dwApplyFlag = (g_DsDefersSettings) ? DS3D_DEFERRED : DS3D_IMMEDIATE;

 g_dsListenerParams.fDopplerFactor = fDopplerFactor;
 g_dsListenerParams.fRolloffFactor = fRolloffFactor;
 if (g_pDSListener)
 g_pDSListener->SetAllParameters (g_dsListenerParams, dwApplyFlag);

 g_dsBufferParams.fMinDistance = fMinDistance;
 g_dsBufferParams.fMaxDistance = fMaxDistance;
 if (g_pDS3DBuffer)
 g_pDS3DBuffer->SetAllParameters (g_dsBufferParams, dwApplyFlag);
}
```

- OOP: Objekte initialisieren, Elementfunktionen aufrufen

Digitale Audioverarbeitung | WS 2000 | 18.205

## DirectSound: 3D-Effekte

Basisfunktionen:

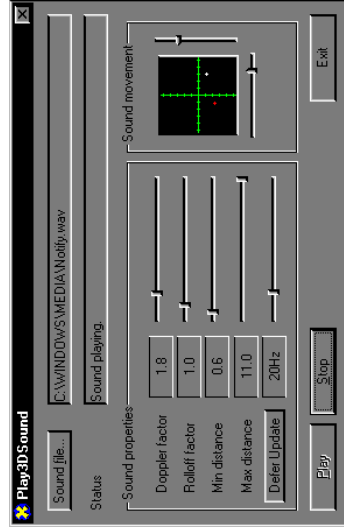
- Position, Geschwindigkeit, Ausrichtung des Hörers
- Position, Geschwindigkeit, Ausrichtung der Schallquelle(n)
- Schallkegel einer Schallquelle

berechnete Effekte:

- Dämpfung (rolloff): Abstand zwischen Quelle und Hörer
- Dopplereffekt: relative Geschwindigkeit Quelle vs. Hörer muß explizit gesetzt werden (wird nicht aus  $dx(x,y,z)/dt$  berechnet)
- Lautsprecherkonfiguration
- HRTF wenn in HW vorhanden, nicht emuliert

Digitale Audioverarbeitung | WS 2000 | 18.205

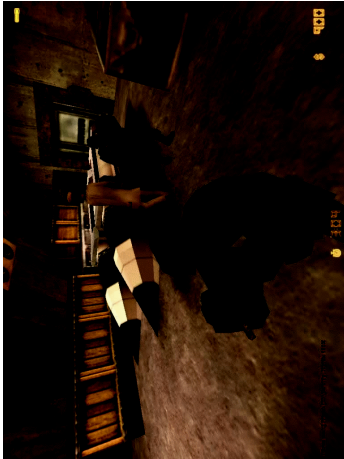
## DirectSound: DX7 SDK Demo



- Beispiel für eine "minimale" DirectSound3D-App.
- lädt und spielt eine .wav-Datei ab
- Effekte interaktiv einstellbar

Digitale Audioverarbeitung | WS 2000 | 18.205

## DirectSound: typ. 3D-Shooter



[counter-strike]

- 3D-Listenobjekt für den Spieler
- 3D-Schallquellen für jeden Gegner (Schritte, Waffen, Gespräche)
- Schallquellen für die Umgebung (Türen, Fabrikgeräusche, usw.)
- Schallquellen für Waffen und Expositionen

Digitale Audioverarbeitung | WS 2000 | 18.205

## EAX

Creative Labs "Environmental Audio Extensions":

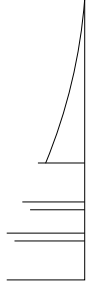
- Erweiterung für DirectSound
- über neue IDirectSound-SubInterfaces
- nutzt Hardware (z.B. Soundblaster live!) oder Software
- Einstellung von Echo- und Halleffekten
  - global für den primären Audiopuffer
  - oder für einzelne DirectSound Sources
  - HW-Unterstützung für occlusion / obstruction
- modelliert die akustische Umgebung
- entscheidende Verbesserung der Audioqualität
- diverse Presets, z.B. "concert hall" oder "bathroom"

Digitale Audioverarbeitung | WS 2000 | 18.205

## EAX: Konzept

- Reflektierter Schall maßgeblich für Raummempfindung
- Raumgröße, Art des Raums, Material der Wände, ...
- auch rein akustisch: "im Dunkeln"
- weitgehend einstellbar über Hall-Parameter:

Verzögerung, Intensität der frühen Reflexionen  
Dichte, Spektrum, Dauer der Hallfahne



- Presets für typische Situationen
- individuell einstellbare Parameter zur Feinkontrolle

Digitale Audioverarbeitung | WS 2000 | 18.205

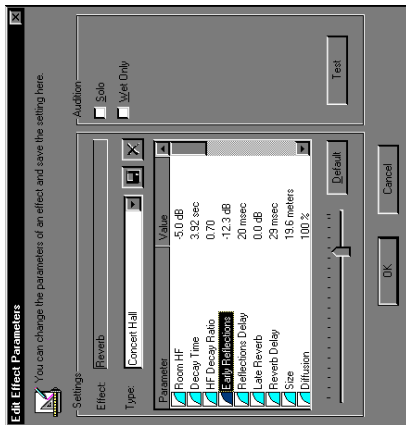
## EAX: Einbindung in DirectSound

- DirectSound-API unterstützt "property sets"
- Anwendung kann die Attribute abfragen und setzen
- EAX als zusätzliche Attribute für IDirectSound:

```
... // declarations omitted
EAXDirectSoundCreate(NULL, &pDirectSoundObj, NULL);
pDirectSoundObj->SetCooperativeLevel(hWnd, DSSCL_EXCLUSIVE);
pDirectSoundObj->CreateSoundBuffer(&desc, &pPrimaryBuf, NULL);
pPrimaryBuf->QueryInterface(IID_IDirectSound3DListener, &pListener);
pDirectSoundObj->CreateSoundBuffer(&desc, &pSecondaryBuf1, NULL);
pSecondaryBuf->QueryInterface(IID_IDirectSound3DBuffer, &p3DBuf1);
...
p3DBuf[0]->QueryInterface(IID_IKsPropertySet, &pEAXListener);
p3DBuf[i]->QueryInterface(IID_IKsPropertySet, &pEAXSource[i]);
...
LONG Reverb = -600;
pEAXListener->Set(DSRPROPSIID_EAX_ListenerProperties,
DSRPROPERTY_EAXLISTENER_REVERB | DSRPROPERTY_EAXLISTENER_DEFERRED,
NULL, 0, &Room, sizeof(LONG));
...
```

Digitale Audioverarbeitung | WS 2000 | 18.205

### EAX: Sblive Hall-Parameter



- Vielfalt vergleichbar mit professionellen Hallgeräten
- geeignete Presets für viele Apps. und Spiele

### EAX: "obstruction"

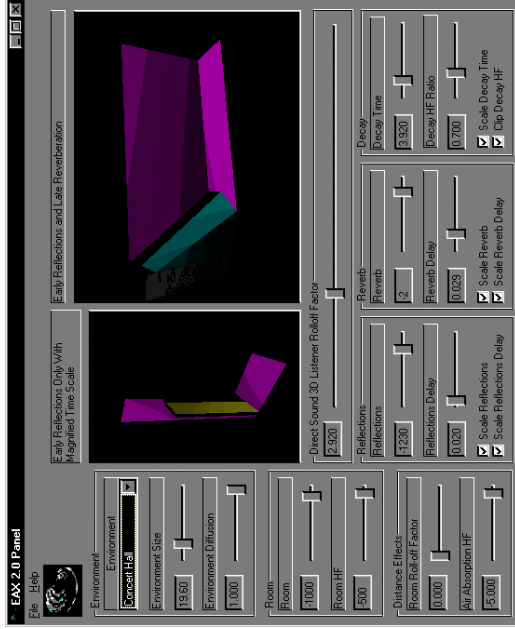
obstruction:

- Quelle hinter Hindernis im selben Raum
- direkter Schall ist gedämpft
- indirekter Schall unverändert

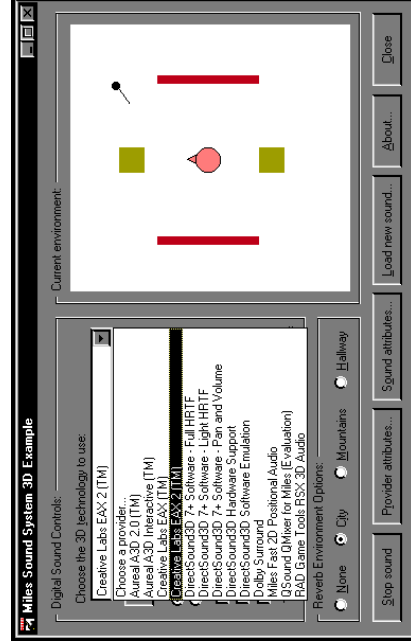
occlusion:

- Schallquelle hinter einer Wand in separatem Raum
- Dämpfung entsprechend Wanddicke und Material
- sowohl direkter als indirekter Schall sind gedämpft
- Realisierung über Tiefpassfilterung
- quantitativ in: IASIG "interactive 3D audio rendering guidelines"

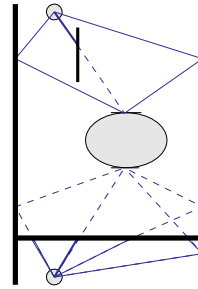
### EAX: SDK Demo



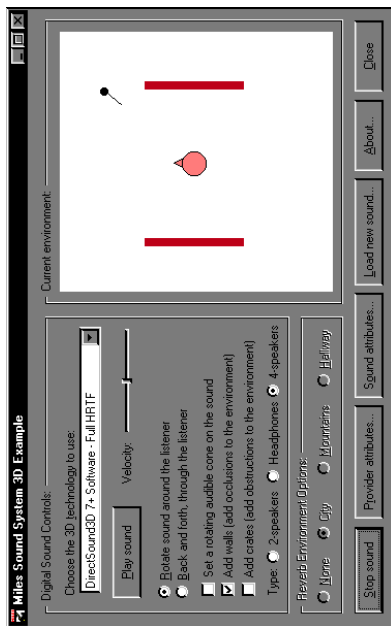
### Miles Sound System: 3D Provider



- [www.smacker.com/mss3d.htm](http://www.smacker.com/mss3d.htm), erlaubt Vergleich der Verfahren:
- DirectSound / A3D / EAX / RSX / ...

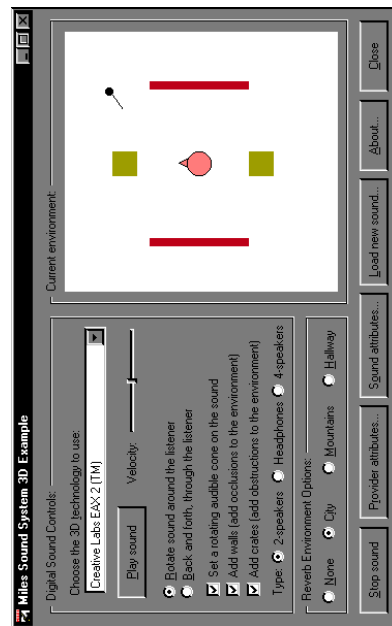


### Miles Sound System: DirectSound + HRTF



- generische HRTF, am besten via Kopfhörer
- Obstruktion ja, aber keine Reverb-Effekte

### Miles Sound System: EAX 2.0



- voller Funktionsumfang von DirectSound
- plus Effekte der Sblivel, inklusive Reverb

### OpenAL



OpenAL: the "open audio library"

- [...] joint effort
- vendor-neutral
- cross-platform
- open API
- for interactive, primarily spatialized audio
- derzeit: Creative Labs, Loki Interactive
- portabler Code
- derzeit: BEOS, Mac, Linux, Windows
- open source, LGPL
- Spezifikation für OpenAL 1.0 im September 2000
- erster Code und CVS unter [www.openal.org](http://www.openal.org) bzw. [cvs.loki.games.com](http://cvs.loki.games.com)

### 3D-Audio: Status, Ende 2000

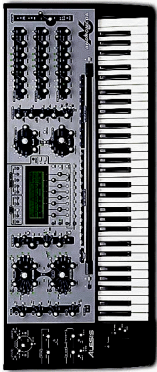
- 3D Musikaufnahmen bisher kaum erhältlich
- SACD und DVD-Audio gerade in der Markteinführung
- Musikaufnahmen zunehmend als "Videoclips" mit AC3-Ton
- Mehrkanalton im Kino unverzichtbar
- auch auf DVDs
- Formate: Dolby Digital, DTS, MPEG-2
- Mehrkanal-Soundkarten etabliert
- z.B. Sblive 5.1 (6 analoge Ausgänge)
- 3D APIs verfügbar
- insbesondere DirectSound + EAX
- für 2 oder 4 Lautsprecher oder Kopfhörer

## Klangsynthese: Agenda

- Motivation
- Literatur
- Sprache vs. Musik

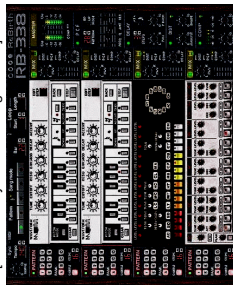
### Verfahren

- subtraktiv: klassische "analoge" Synthese
- Wavetable-, Sampling
- additiv
- Frequenzmodulation
- physical modeling
- Granularsynthese
- Mischformen



[Alesis Andromeda]

[Software-Synth: Steinberg ReBirth]



## Klangsynthese: Literatur

- E.R. Miranda, Computer sound synthesis for the electronic musician, LEM A MIR 26466  
 M.J. Manthey, the DGG real-time sound synthesizer, DAIMI PB-56 1976, R5522  
 C. Chapman, formant musik-synthesizer, elektor Verlag, 1978

SMS spectral modeling: <http://www.iaa.upf.es/~sms/>  
 Synth FAQ: <http://titl.largo.fi.us/faq/synthfull.html>

[www.synthmuseum.com](http://www.synthmuseum.com)  
[www.synthzone.com](http://www.synthzone.com) (aktuelle und historische Synthesizer)  
[www.hammersound.net](http://www.hammersound.net) (free SoundFont library)

Sprache: Links unter <http://www.colli.uni-sb.de/phonetik/projects/bookmarks.html>

## Klangsynthese: Literatur

Synth FAQ: <http://titl.largo.fi.us/faq/synthfull.html>

SMS spectral modeling: <http://www.iaa.upf.es/~sms/>

[www.alesis.com/](http://www.alesis.com/) (Andromeda, AirFX )  
[www.bigbriar.com](http://www.bigbriar.com) (Bob Moog)  
[www.steinberg.de](http://www.steinberg.de) (Cubase VST Homepage)  
[www.native-instruments.de](http://www.native-instruments.de) (Software-Synthesizer)  
[www.keyboards.de/soundforum](http://www.keyboards.de/soundforum) (Gratis-Software-Synth zur Serie)  
[www.nemesysmusic.com](http://www.nemesysmusic.com) (Gigasampler)

[www.webspeech.de](http://www.webspeech.de) (Logox)  
[http://www.colli.uni-sb.de/phonetik/projects/Sprgs\\_pu.html](http://www.colli.uni-sb.de/phonetik/projects/Sprgs_pu.html) (Logox publications)

## Klangsynthese: Ziele

- Wiedergabe kodierter / komprimierter Daten
- Reproduktion von Musikinstrumenten
  - leichter spielbar
  - billiger, portabel, flexibler, leiser/lauter, ...
  - Rekonstruktion defekter, verlorener Instrumente
- neuartige Klänge
  - Vielfalt der möglichen Algorithmen
  - Erinnerung: Wahrnehmung: Grundton plus Spektrum
- Sprachsynthese



## Musiksynthese: Klassifikation

musikalische Gesichtspunkte:

- Eingabegeräte
- Spieltechniken und -hilfen
- Klangmöglichkeiten
- Echtzeitfähigkeit, Stimmenzahl

technische Gesichtspunkte:

- Syntheseverfahren (subtraktiv, FM, ...)
- Implementierung (analog, digital)
- uva.

Digitale Audioverarbeitung | WS 2000 | 18.205

## Musiksynthese: Eingabegeräte



[L. Theremin 1927]



[Alesis airFX 2000]

- Tasteninstrumente (Klaviiatur)
- Sampler, virtuelle Instrumente (Software: MIDI)
- Theremin, "e-beam", "airFX" (berührungslos)
- E-Drums (Schlagzeug)
- Gitarrensynthesizer (gezupfte Saiten)
- E-Blasinstrumente (Blaswandler)
- Vocoder (Stimme)
- Sprachsynthese (Textdaten)

=> entsprechend vielfältige Anforderungen

Digitale Audioverarbeitung | WS 2000 | 18.205

## Musiksynthese: Trends

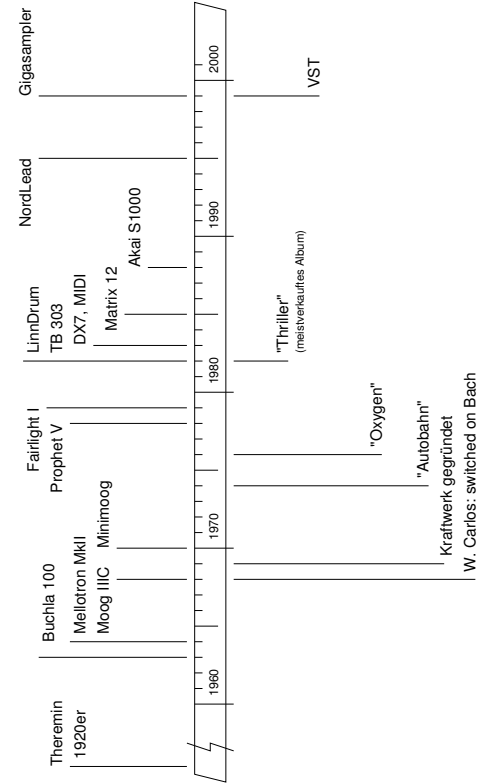
zunehmende Rechenleistung für DSPs / PCs:

- virtuell analoge Synthesizer
- modulare Synthesizer
- integrierte Effekte
- virtuelle Instrumente (z.B. VST Plugins)
- echtzeitfähige Klangveränderungen
- mehrkanalfähige Geräte (5.1)
- 96 KHz, 24 bit
- photorealistiche Oberflächen (G-Media M-Tron mit Kaffeeflecken...)



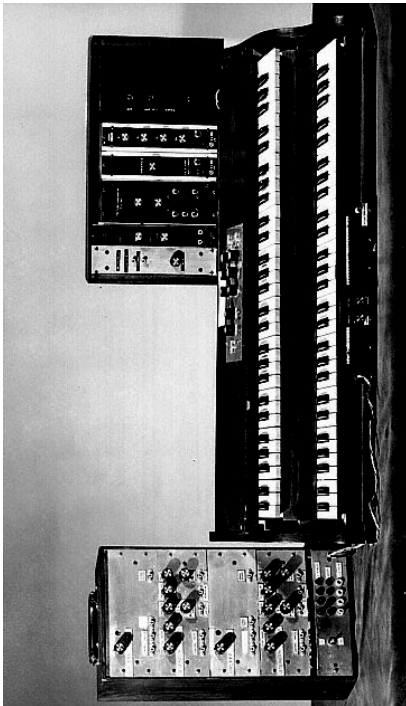
Digitale Audioverarbeitung | WS 2000 | 18.205

## Musiksynthese: Zeitleiste



Digitale Audioverarbeitung | WS 2000 | 18.205

## Analoge Synthese: Moog #1

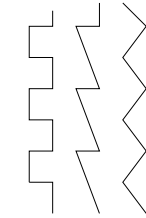


- gebaut mit/für Herbert Deutsch, Juli bis September 1964

Digitale Audioverarbeitung | WS 2000 | 18.205

## Analoge Synthese: Konzept

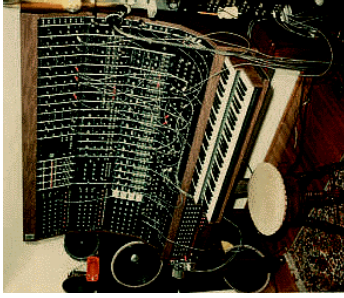
- additive Synthese wegen Kosten unmöglich
- subtraktive Synthese:
- obertonreiches Ausgangssignal
  - gewünschten Klang durch Filterung erzeugen
  - zusätzliche Modulation für "lebendigen" Klang



- Oszillatoren mit obertonreichen Klängen:
  - Rechteck, PWM Rechteck, Impulse
  - Sägezahn, Dreieck
  - Rauschen
  - oder unreiner Sinus
- musikalisch sinnvolle Filter, vor allem Tiefpaß
- Anschlagdynamik, Hüllkurven, Unisono, ...

Digitale Audioverarbeitung | WS 2000 | 18.205

## Analoge Modularsysteme



[www.moogarchives.com/rfmod2.htm]

- Vielzahl einzelner Funktionsblöcke:
- Oszillatoren (div. Signalformen)
  - div. Filter
  - Verstärker
  - Hüllkurvengeneratoren
  - spez. Module, z.B. Ringmodulator, usw.

"patches": Einstellung (und Speicherung) von Klängen erfordert:

- Konfiguration aller Verbindungskabel
- Schalter- / Potentiometerstellungen

Digitale Audioverarbeitung | WS 2000 | 18.205

## Analoge Synthese: Minimoog

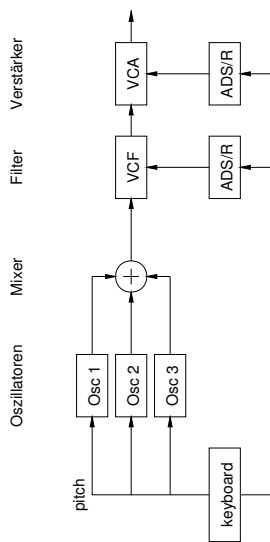


[Minimoog Model D, 1970]

- fester Schaltungsaufbau mit VCO, VCF, VCA plus LFO
- spannungsgesteuerte Komponenten: 1 V / Oktave
- monophon, drei Oszillatoren
- Tiefpaßfilter mit 12/24 db/Oktave
- legendärer Klang

Digitale Audioverarbeitung | WS 2000 | 18.205

### Minimoog: Blockschaltbild



einfacher, aber ausreichend flexibler Signallauf:

- Vorbild für praktisch alle späteren Synthesizer
- Erweiterungen über flexiblere Oszillatoren
- und weitere Modulationsmöglichkeiten

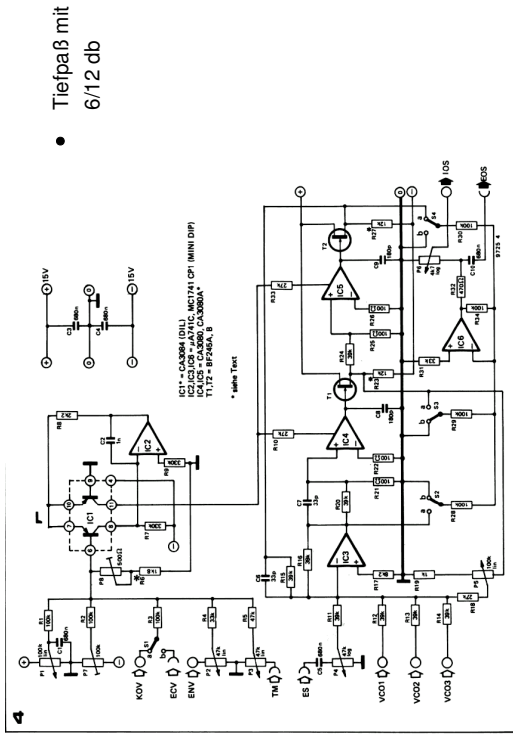
### Formant

- einfacher subtraktiver Synthesizer
- als Selbstbauprojekt der Zeitschrift "elektor", 1978 [C. Chapman]
- "mit erträglichem Klang" (P. Gorges, keyboards 01/01)
- patentfreie Schaltungen für VCO/VCF/VCA
- sehr gut dokumentiert

durchaus interessante Lektüre:

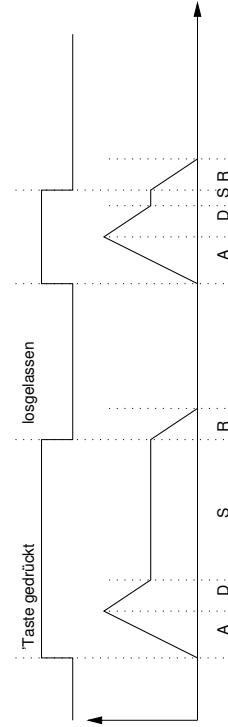
- Tips für typische Parametereinstellungen
- Schaltungsbeispiele: Dreieck->Sinus (absichtlich kein guter Sinus)

### Formant: VCF

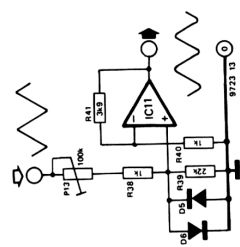


- Tiefpaß mit 6/12 db

### Hüllkurven: ADSR



- attack      Anschlagphase      Zeit
- decay      Abfallzeit nach dem Anschlag      Zeit
- sustain      eigentlicher Klang      Level
- release      Ausklingen      Zeit
- bei Bedarf mehrstufige / komplexere Hüllkurven
- attack/decay-Phase besonders wichtig für den Klang



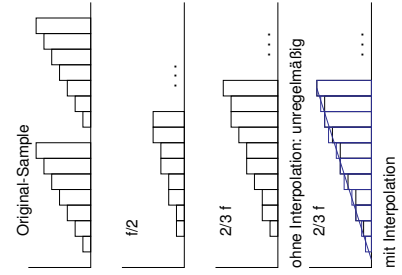


## Sampling

- Aufzeichnen von natürlichen Klängen / Geräuschen
  - getriggerte Wiedergabe dieser "samples"
- in Verbindung mit einem subtraktiven Synthesizer:
- Verwendung der Samples als Ausgangsmaterial
  - statt der "einfachen" Oszillatoren
  - erfordert ein/mehrere Samples pro Ton
  - oder Frequenzanpassung (pitch-shifting) der Samples
  - Klangformung via Filter und Verstärker
  - Hüllkurven, Anschlagdynamik, usw.
  - fast immer digital realisiert (Ausnahme: Mellotron)
  - zusätzlich Digitaleffekte

Digitale Audioverarbeitung | WS 2000 | 18.205

## Sampling: Pitch-Shifting

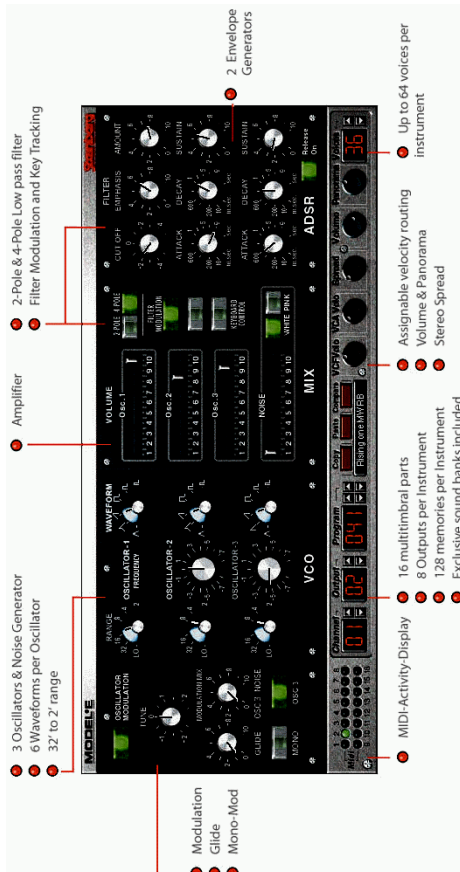


- Speicher ist teuer
  - ca. 75 KB pro Sekunde Sample
- => Sample für mehrere Tonhöhen nutzen
- $OSC[i] = SAMPLE[f/i0 \cdot i]$
- triviale Realisierung unbrauchbar
  - daher Interpolation der Samplewerte
  - z.B. 8 Punkte-Interpolation (E-mu)

- aber Aliasing-Probleme:
- Up-Shifting verletzt das Abtasttheorem
  - Überlagerung mehrerer Stimmen problematisch

Digitale Audioverarbeitung | WS 2000 | 18.205

## virtuell analog: Steinberg Model\*E



Digitale Audioverarbeitung | WS 2000 | 18.205

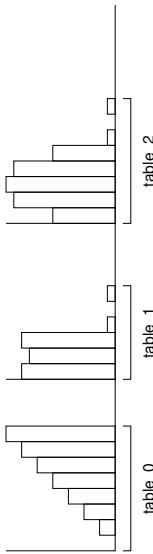
## virtuell analog: Demos

- Klangbeispiele:
- Nord Modular Demo
  - Nord competition / 2000
  - Access Virus Demo
  - keyboards 1x/2000
  - Steinberg Model\*E Demo
  - [www.steinberg.de/](http://www.steinberg.de/)

- Konzept ist weitgehend ausgereizt
- höhere Rechenleistung ermöglicht weiter steigende Qualität
- bereits alle "relevanten" Klassiker nachgebaut
- Modularsysteme für bel. komplexe Setups

Digitale Audioverarbeitung | WS 2000 | 18.205

### Sampling: "Wavetable"

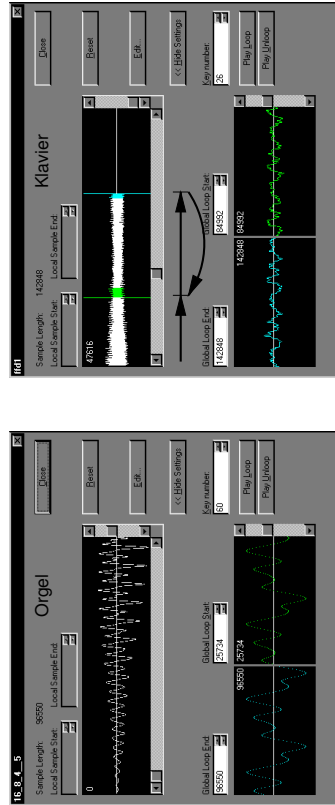


$$OSC[i] = WAVETABLE[i/10 * j]$$

#### Wavetable-Synthese:

- subtraktive Synthese
- kurze synthetische Wellenformen
- als Oszillatoren (mit pitch-shifting)
- oft als Synonym für Sampling benutzt

### Sampling: Loops



- wiederholte Wiedergabe eines Samplebereichs
- um Speicherplatz zu sparen: Attack + Loop + (opt) Release
- Länge und Qualität der Loops maßgeblich für Klang
- spez. Tools zum Finden / Erzwingen von Looppunkten

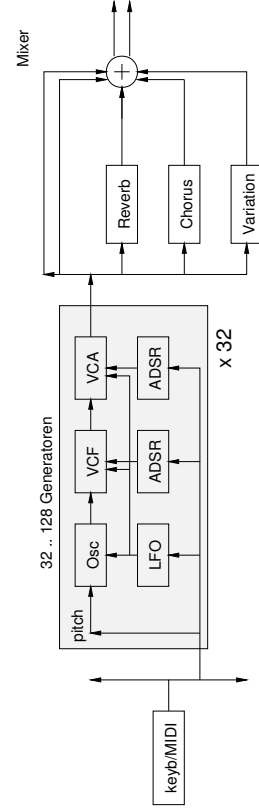
### Sampling: Klangformung

möglichst natürlicher bzw. differenzierter Klang:

- Verwendung möglichst vieler und langer Samples
- Hüllkurven für Filter und Verstärker
- "velocity layers": mehrere Samples pro Ton
- Modulation via LFO
- Samples für mehrere Spieltechniken (gezupfte vs. gestrichene Saite, usw.)
- Beispiel zur Auswahl der Samples:

Gigasampler Instrument Zones

### Sampling: Hardware



typische Daten für aktuelle "ROMpler":

- 32 .. 128 stimmig, 48 KHz, 16 bit
- 16 .. 64 MB ROM, 16 .. 256 MB RAM (Sampler)
- mehrere LFOs, separate Hüllkurven für VCF und VCA
- inkl. Mischer, plus Digitaleffekte

## Sampling: Gigasampler

- Looping oft nicht akzeptabel (insb. für klassische Instrumente)
  - Kosten proportional zum Speicherausbau
  - RAM deutlich teurer als Festplatten
- =>
- Gigasampler: [www.nemesismusic.com](http://www.nemesismusic.com)
  - Samplestarts im RAM
  - spätere Samplewerte nur bei Bedarf laden
  - praktisch keine Größenbeschränkung der Samples
  - z.B. 1 GB Klavier

Zugriffszeit HD: 10 ms  
 Transferrate: 5 MB/s

Puffer pro Sample: mind. 480 Werte, ~ 1KB  
 Stimmenanzahl: 5 MB/48K ~ 100

## Sampling: SoundFont 2

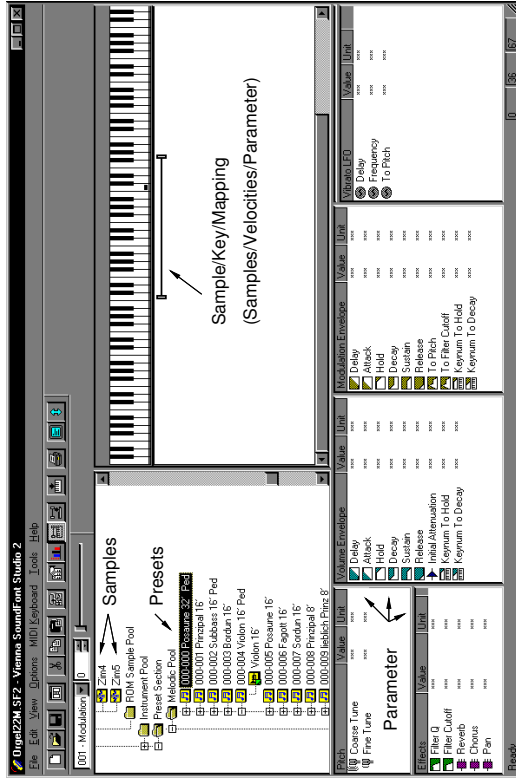
- Dateiformat für Samplebänke, Creative Labs 1997
- ursprünglich für Soundblaster 64 AWE
- RIFF-Struktur, insgesamt 22 Chunk-Typen:

```

RIFF
 SFBK
 INFO-list
 header chunks: name, version, ...
 sdta-list
 sample data
 pdta-list
 presets (headers, list)
 instruments (headers, list)
 sample headers

```

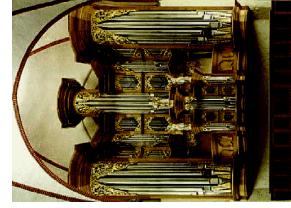
## SoundFont: Vienna Editor



## additive Synthese

Klangerzeugung durch Überlagerung

- der Grundtons
- und aller einzelnen Obertöne
- benötigt N Klangquellen / Oszillatoren
- und zugehörige Regelung, z.B. Hüllkurven
- entsprechender Hardwareaufwand
- dafür im Prinzip keine Filter erforderlich
- Kirchenorgel, z.B. 60 Register a 60 Töne: 3600+ Pfeifen
- Hammondorgel, 91 tonewheels, elektronische Verstärkung
- diverse Orgel/Hammondorgel-Nachbildungen
- ansonsten additive Synths. sehr selten

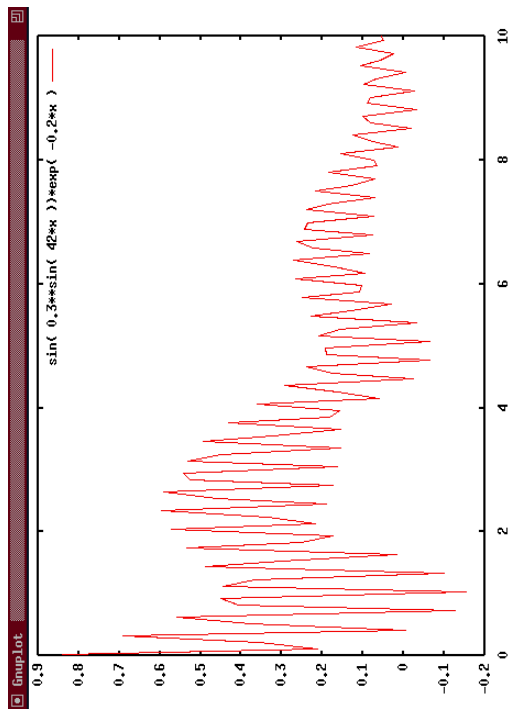


[Amp Schmitz Orgel, St. Jacobi HH]

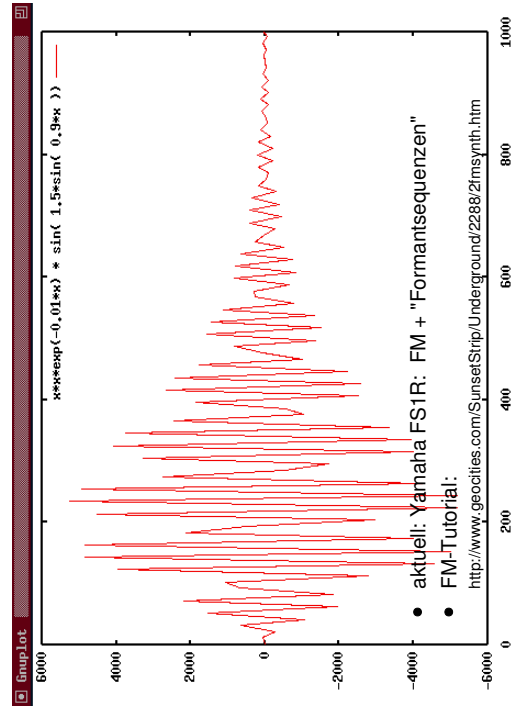




### Frequenzmodulation: Beispiel



### Frequenzmodulation:



### Physical Modeling

- warum überhaupt vereinfachte Syntheseverfahren?
- "physical modeling synthesis"
- Modellierung realer / imaginärer Schallquellen
- als physikalische Systeme: Saiten, Pfeifen, Resonanzkörper, ...
- Aufstellen der Bewegungsgleichungen
- numerische Lösung der Bewegungsgleichungen
- möglichst in Echtzeit
- im Prinzip beliebig genaue Simulation jedes Instruments
- aber extremer Rechenaufwand

### Physical Modeling: Prinzip

```

/***** PIUCK C *****/
C-Simulation fortschreitender Wellen (traveling waves) zur
Erzeugung von Wellenformen, (nach Helmholtz 1857, Cremer 1979
und J.O. Smith 1992), speicherbar als ASCII- oder .WAV-Datei.
Demo-Programm fuer eine Waveguide-Implementation (physical
modeling): Anzupfen einer Gitarren-Saite.
Dr. Justus Noll c't 07/95 Seite 240ff
*****/

int initWelle (double Amplitude, /* z.B. 0.5 */
double Frequenz, /* z.B. 100 [Hz] */
double ZupfStelle, /* z.B. 0.1 (Saitenlaenge) */
double SampleOffset /* z.B. 22 */
)
{
 int i;
 int LeitungsLaenge, StartSample;
 double *AnfangsForm, *anflanke, *abflanke;

 LeitungsLaenge=SAMPLERATE/Frequenz/2+1;
 StartSample=MAXSAMPLE=max(LeitungsLaenge*ZupfStelle,1);
 AnfangsForm=(double*) calloc(LeitungsLaenge, sizeof(double));
 if (AnfangsForm==NULL) Fehler(3);
 anflanke=Amplitude/StartSample;
 abflanke=Amplitude/(LeitungsLaenge-StartSample-1);
 for (i=0;i<StartSample;i++) *AnfangsForm[i]=anflanke*i;
 for (i=StartSample;i<LeitungsLaenge;i++)
 AnfangsForm[i]=abflanke(LeitungsLaenge-1-i);
 ULEITUNG
 =VerzLeitungAn(ULEITUNG,AnfangsForm,0.5,LeitungsLaenge);
 OLEITUNG
 =VerzLeitungAn(OLEITUNG,AnfangsForm,0.5,LeitungsLaenge);
 return SampleOffset*LeitungsLaenge;
}
...

```

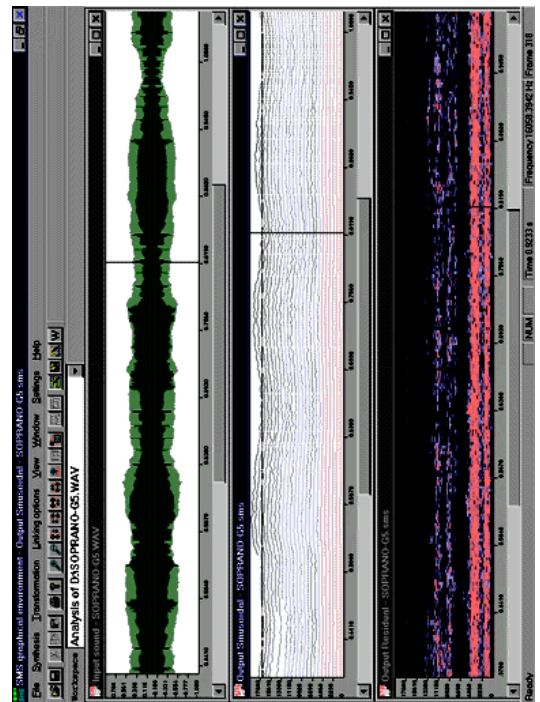


## Spectral Modeling

- kombiniertes Analyse- und Syntheseverfahren
- X. Serra, Barcelona, seit ca. 1989: [www.iaa.ept.es/sms/](http://www.iaa.ept.es/sms/)
- Fourieranalyse der Eingangssignale
  - Erkennen der dominanten Spektrallinien
  - und deren zeitlichen Verläufen
  - Interpretation des rechtlichen Spektrums als Rauschen
  - Abspeichern des zeitlichen Verlaufs
- additive Synthese der erkannten Spektrallinien
- und Hinzufügen des originalen Rauschens
- erlaubt sehr authentische Klangmodifikationen
- Klangbeispiele

Digitale Audioverarbeitung | WS 2000 | 18.205

## Spectral Modeling:



Digitale Audioverarbeitung | WS 2000 | 18.205

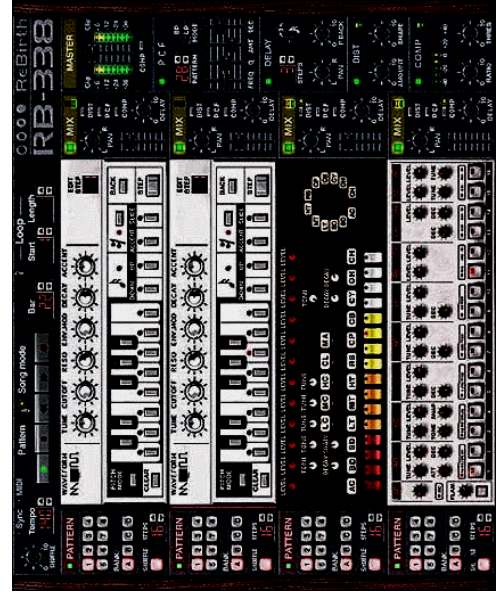
## Drum Synths

Nachbildung von Schlagzeug-Sounds:

- analog, z.B. Roland TR-808 und TR-909
- digital per Sampling: Linn Drum ... Alesis Dm pro
- keine besonderen Anforderungen an das Sampling:
  - kurze perkussive Klänge, keine Loops notwendig
  - kein pitch-shifting notwendig
  - bei Bedarf Alternativ-Samples / velocity-mapping
- typische ROMpler: ~40 Drum-Sets (je ca. 40 Klänge)
- große Vielfalt an Geräten
- Beispiel Rebirth, [www.propellerheads.se](http://www.propellerheads.se)

Digitale Audioverarbeitung | WS 2000 | 18.205

## Drums: Rebirth



Demoversion unter [www.propellerheads.de](http://www.propellerheads.de)

Digitale Audioverarbeitung | WS 2000 | 18.205



### MIDI: Literatur

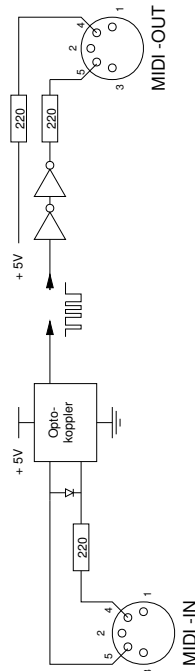
- The complete MIDI 1.0 Detailed Specification, MIDI Manufacturers Association, 1996  
[www.midi.org](http://www.midi.org)
- XG (Europe) homepage, [www.yamaha.co.uk/xg/](http://www.yamaha.co.uk/xg/)
- XG specification and guidelines, Yamaha Inc., [www.yamaha.co.uk/xg/](http://www.yamaha.co.uk/xg/)
- MU100R toneregulator, sound list and MIDI data, Yamaha, 1997
- Cubase VST Einführung (und Docs), Steinberg GmbH, Hamburg, 2000
- VST2 specification, Steinberg GmbH, Hamburg, 2000
- Cubase VST homepage, [www.steinberg.de](http://www.steinberg.de)
- [xgmidi.wral.de](http://xgmidi.wral.de)
- [www.yme.co.uk/yeme/xgedit.htm](http://www.yme.co.uk/yeme/xgedit.htm) (XGedit)
- [www.mhv-online.de](http://www.mhv-online.de) (Midi File Hersteller, Deutschland)

### MIDI: Geschichte

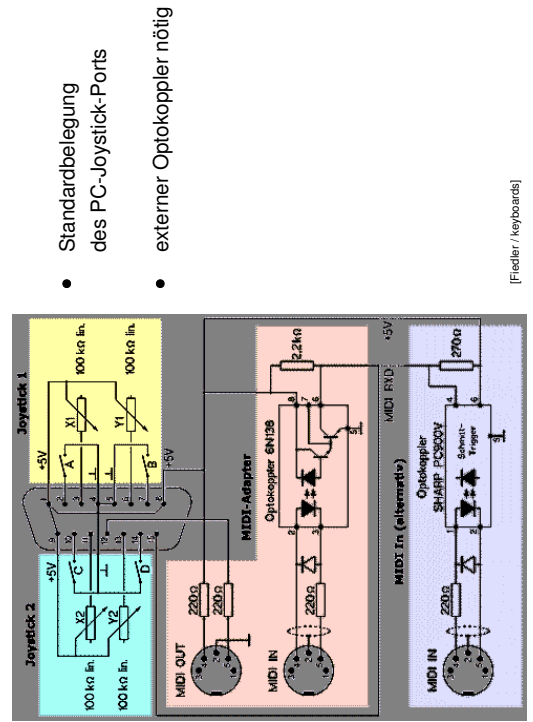
- 1981 erste Vorarbeiten zur Kopplung von Synthesizern
- 1983 erste Gerätekopplung auf der NAMM:  
Sequential Circuits Prophet 600 - Roland Jupiter 6
- 1988 Standard MIDI Files
- 1991 General MIDI
- 1992 MIDI machine control
- 1994 Yamaha XG Spezifikation
- 1998 VST
- 1999 Downloadable Sounds DLS 1/2
- 2001 MIDI over 1394

### MIDI: Hardware

- digitale Übertragung
- Protokoll wie RS232, 8N1 (logical 0 = current on)
- feste Bitrate von 31.250 baud
- Potentialtrennung (Vermeidung von Brummschleifen)
- unidirektional: bei Bedarf zwei Leitungen
- MIDI IN / OUT / THRU
- MIDI THRU liefert Kopie des MIDI IN-Signals

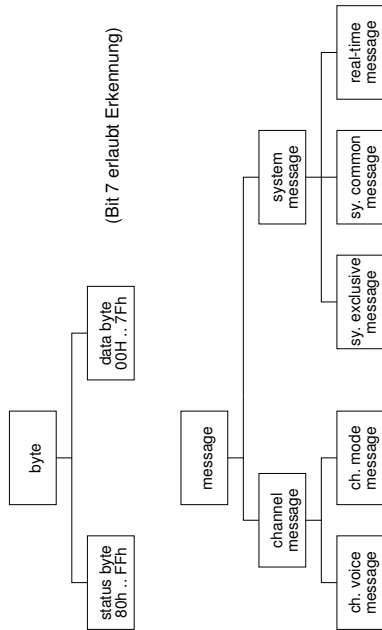


### MIDI: Soundblaster / PC



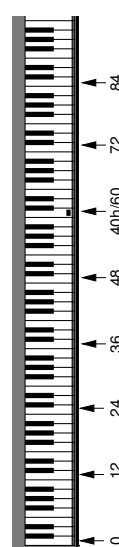
- Standardbelegung des PC-Joystick-Ports
- externer Optokoppler nötig

### MIDI: Datentypen



- Format: status (command) byte, data byte(s)
- 16 Kanäle (channels) auf einem Kabel

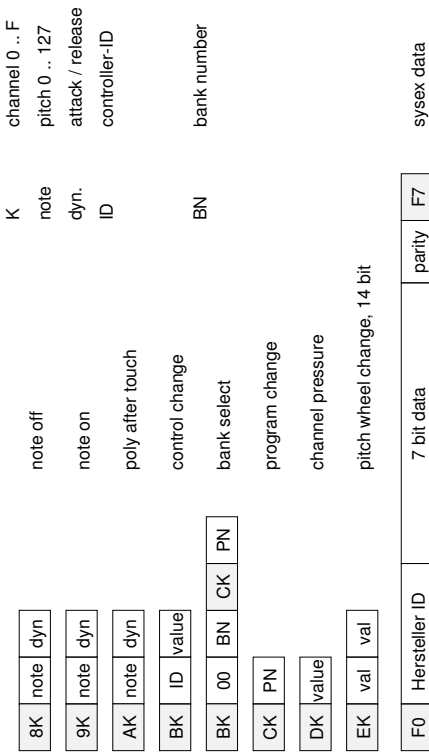
### MIDI: Tonhöhe



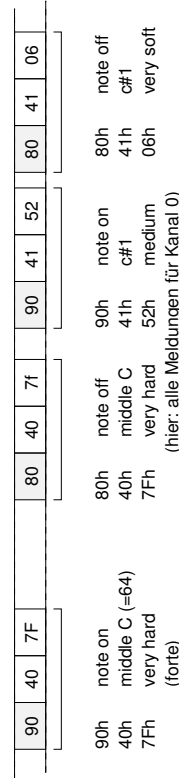
Konzept der Tonhöhe (pitch):

- Tasteninstrument mit festen Tonhöhen
- wohltemperierte Stimmung
- kodiert als ein Byte, 0 .. 127
- "Schlüssel-C" = 60, C# = 61, D = 62 usw.
- andere Skalen möglich: "tuning"-Meldungen
- Feintuning via "pitch wheel" Meldungen (14 bit Auflösung)

### MIDI: Channel messages



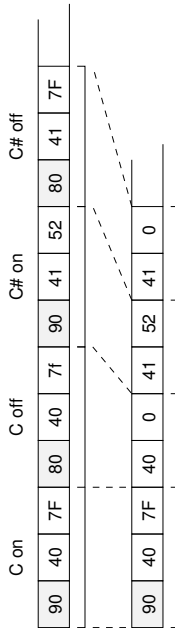
### MIDI: note on, note off



"keyboard"-orientiert:

- feste Tonhöhen
- Anschlagstärke (attack/release)
- Modulation via "controllers"
- "all notes off" für Paniksituationen

### MIDI: running status



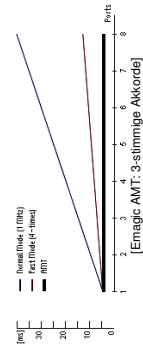
- niedrige Bandbreite des MIDI-Kanals
  - Befehlsbytes möglichst einsparen
- => "running status"
- Beispiele:
- "note off" durch "note on 0" ersetzen
  - "control changes" oder "pitch change" messages

### MIDI: Bandbreite

- 31kbps: ein Byte in 0.3 ms => "note on" dauert 1 ms
- exakt taktgenaue Übertragung (z.B. Akkorde) unmöglich

- zeitkritische Signale (Perkussion) zuerst senden
- mehrere Kanäle parallel verwenden
- Ansteuerung mit Zeitmarken

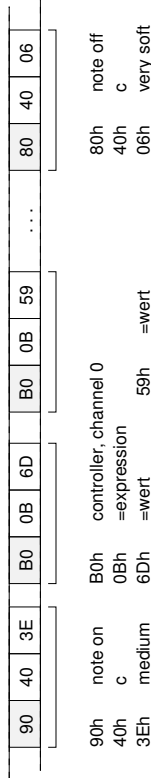
z.B. Emagic "active midi transmission"



Yamaha mLAN: Netzwerk für Audio/MIDI/Video

- nutzt IEEE 1394 Bus ("FireWire")
- Datendurchsatz bis 400 Mbps
- bis 100 Audiokanäle und 256 MIDI-Kanäle parallel
- [www.yamaha.co.jp/english/mLAN/index.htm](http://www.yamaha.co.jp/english/mLAN/index.htm)

### MIDI: controller change



diverse "controllers" fest definiert

- "continuous" / "switch"
- Feineinstellung (MSB+LSB)
- Beispiele: s.u.

| Nummer     | Bedeutung             |
|------------|-----------------------|
| 0 .. 31    | MSB continuous        |
| 32 .. 63   | LSB continuous 0 ..31 |
| 64 .. 95   | single-byte           |
| 96 .. 101  | incr/decrement        |
| 101 .. 119 | undefined single-byte |

### MIDI: controllers (Auswahl)

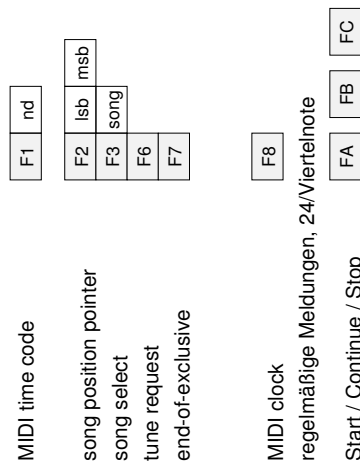
| Controller # | Funktion               | Wertebereich                  |
|--------------|------------------------|-------------------------------|
| 0 / 32       | bank select modulation | 0 .. 127 / 0 .. 127           |
| 1 / 33       | breath controller      | 0 .. 127                      |
| 2            | foot controller        | 0 .. 127                      |
| 4            | portamento time        | 0 .. 127                      |
| 5            | RPN/NRPN data          | 0 .. 127                      |
| 6 / 38       | main volume            | 0 .. 127                      |
| 7            | pan position           | 0 .. 63 = off, 64 .. 127 = on |
| 10           | expression (volume)    | 0 .. 63 = off, 64 .. 127 = on |
| 11           | sustain pedal on/off   | 0 .. 127 (= -64 .. +63)       |
| 64           | portamento             |                               |
| 65           | attack time            |                               |
| 73           |                        |                               |

### MIDI: channel mode messages

Mode-Befehle:

- Umschaltung globaler Parameter
  - omni on / off Aktivierung aller 16 / bestimmter Kanäle
  - poly on / off mehrstimmiges Spiel
  - reset all controllers Initialisierung
  - all sound off (aktueller Kanal)
  - all notes off (alle Kanäle)
  - local control eigenes Keyboard aktivieren?

### MIDI: system common messages



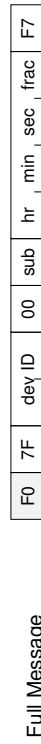
- Details siehe MIDI specification

### MIDI: time code

- Synchronisation mehrerer Geräte?
- insbesondere auch Synchronisation Audio / Video für Film

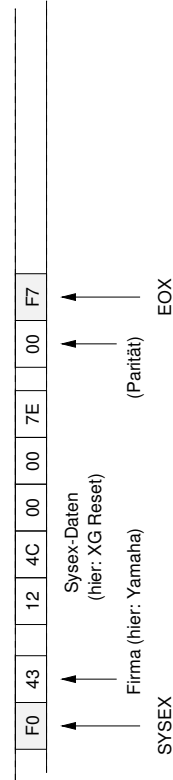


- regelmäßige Übertragung kurzer (relativer) Zeitmarken
- alle 1/30 Sekunde
- F1 0nnn dddd: 0/1 frame (LSB/MSB), 2/3 sec, 4/5 mins, 6/7 hours
- acht Meldungen nötig zur Dekodierung



- für "fast forward", "rewind", etc.
- enthält komplette Systemzeit

### MIDI: sysex messages



- erlaubt flexible Erweiterungen
- Registrierung einer Hersteller-ID notwendig
- wegen Header-Overhead schlecht für Echtzeit geeignet
- unbegrenzte Länge der sysex-Daten
- andere Geräte warten einfach auf das F7-Schlußbyte
- Erweiterung zum "MIDI sample/file dump" (sehr langsam)



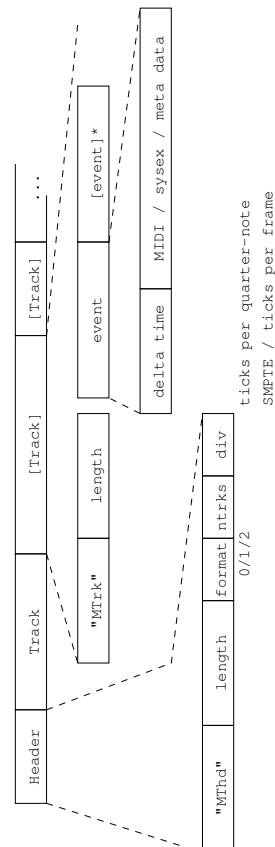
### MIDI: sysex IDs

einige Hersteller-Codes:

|     |                     |
|-----|---------------------|
| 01h | Sequential Circuits |
| 02h | IDP                 |
| 04h | Moog                |
| 24h | Hohner              |
| 40h | Kawai               |
| 41h | Roland              |
| 42h | Korg                |
| 43h | Yamaha              |
| ... |                     |

weitere Hersteller via Bytefolgen: 00h IDh IDh: 00h 00h 01h = Time Warner

### SMF: Standard Midi Files



- chunk-basiertes Dateiformat für MIDI-Daten
  - Tracks enthalten Paare aus (relativen) Zeitangaben und Events
- Typ 0 Header, ein Track mit allen Daten  
 Typ 1 Header, je ein Track pro MIDI-Kanal (Instrument)  
 Typ 2 Header, je ein Track pro "pattern"

### SMF: delta times

- möglichst kompakte Zahlendarstellung der Zeitabstände
- variante Länge, je 7 Nutzbits pro Byte, letztes Byte hat Bit 7 clear

SMF "delta times":

```

0000 0000
0000 0040
0000 007F
0000 0080
0000 2000
0000 3FFF
0000 4000
0010 0000
001F FFFF
0800 0000
0FFF FFFF

```

### SMF: Beispiel

```

4D 54 68 64
00 00 00 06
00 00
00 01
00 60

"MTkd" header
header data length: 6 bytes
format: type 0
ntrks: 1 track
division: 96 per quarter note

4D 54 72 6B
00 00 00 3B
FF 58 94 94 92 18 08
FF 51 03 07 A1 20
00 C0 05
00 C1 2E
00 C2 46
00 92 30 60
00 3C 60
60 91 43 40
...
delta times

```

## General MIDI

- Geräte seit ca. 1990 mehrstimmig und multitimbral
  - ein Gerät reicht als Tonerzeuger für alle Instrumente
- General MIDI Spezifikation (1991):
- Synthesizer mit > 24 Stimmen, 16 fach multitimbral
  - festes Instrumenten-Mapping, 0=Plano, 25=Akustikgitarre, ...
  - ein GM Drumset mit festem Drum-Mapping
  - Kanal 10 reserviert für Drums
  - erlaubt "portable" Kodierung und Wiedergabe von Pop-Songs
  - Roland "SoundCanvas"-Geräte als quasi-Standard
  - aber fragwürdige Qualität auf low-end Geräten
  - Markt für "Midi-Files" mit aktuellen Hits, z.B. [www.hitbit.de](http://www.hitbit.de)

## General MIDI: Instrumente

| Inst. # | Name                   | Inst. # | Name              | Inst. # | Name                 | Inst. # | Name                    |
|---------|------------------------|---------|-------------------|---------|----------------------|---------|-------------------------|
| 1       | Acoustic Grand Piano   | 9       | Clarinete         | 17      | Drumbrg Organ        | 25      | Acoustic Guitar (nylon) |
| 2       | Bright Acoustic Piano  | 10      | Glockenspiel      | 18      | Percussive Organ     | 26      | Acoustic Guitar (steel) |
| 3       | Electric Grand Piano   | 11      | Musik Box         | 19      | Rock Organ           | 27      | Electric Guitar (jazz)  |
| 4       | Honky-Tonk Piano       | 12      | Vibraphone        | 20      | Church Organ         | 28      | Electric Guitar (lead)  |
| 5       | Electric Piano 1       | 13      | Marimba           | 21      | Reed Organ           | 29      | Electric Guitar (clean) |
| 6       | Electric Piano 2       | 14      | Xylophone         | 22      | Accordion            | 30      | Overdriven Guitar       |
| 7       | Harpischord            | 15      | Tubular Bells     | 23      | Harmonica            | 31      | Distortion Guitar       |
| 8       | Clavi                  | 16      | Dulcimer          | 24      | Tango Accordion      | 32      | Guitar Harmonics        |
| 33      | Acoustic Bass          | 41      | Violin            | 46      | String Ensemble 1    | 57      | Trumpet                 |
| 34      | Electric Bass (finger) | 42      | Viola             | 50      | String Ensemble 2    | 58      | Trombone                |
| 35      | Electric Bass (pick)   | 43      | Cello             | 51      | SynthStrings 1       | 60      | Muted Trumpet           |
| 36      | Frei Bass              | 44      | Contra Bass       | 52      | SynthStrings 2       | 61      | French Horn             |
| 37      | Slap Bass 1            | 45      | Tremolo Strings   | 53      | Choir Aahs           | 62      | Bass Section            |
| 38      | Slap Bass 2            | 46      | Pizzicato Strings | 54      | Voice Oohs           | 63      | SynthBass 1             |
| 39      | Synth Bass 1           | 47      | Orchestral Harp   | 55      | Synth Voice          | 64      | SynthBass 2             |
| 40      | Synth Bass 2           | 48      | Timpani           | 81      | Lead 1 (square)      | 90      | Piast 2 (warm)          |
| 65      | Soprano Sax            | 73      | Piccolo           | 82      | Lead 2 (sawtooth)    | 91      | Piast 3 (polysynth)     |
| 66      | Alto Sax               | 74      | Flute             | 83      | Lead 3 (calligae)    | 92      | Piast 4 (choir)         |
| 67      | Tenor Sax              | 75      | Recorder          | 84      | Lead 4 (chiff)       | 93      | Piast 5 (bowed)         |
| 68      | Baritone Sax           | 76      | Pan Flute         | 85      | Lead 5 (chiarung)    | 94      | Piast 6 (metallic)      |
| 69      | Clarinet               | 77      | Bowm Balle        | 86      | Lead 6 (voice)       | 95      | Piast 7 (tuba)          |
| 70      | English Horn           | 78      | Shakunachi        | 87      | Lead 7 (liths)       | 96      | Piast 8 (sweep)         |
| 71      | Bassoon                | 79      | Whistle           | 88      | Lead 8 (bass + lead) | 121     | Guitar Fret Noise       |
| 72      | Clarinet               | 80      | Cocaina           | 89      | Lead 9 (bass)        | 122     | Stream Noise            |
| 87      | FX 1 (rain)            | 105     | Sitar             | 113     | Triangle Bell        | 123     | Shaver Noise            |
| 88      | FX 2 (soundtrack)      | 106     | Banjo             | 114     | Agogo                | 124     | Reverse Cymbal          |
| 89      | FX 3 (cymbals)         | 107     | Shamisen          | 115     | Chimes               | 125     | Reverse Snare           |
| 100     | FX 4 (tom-toms)        | 108     | Shamisen          | 116     | Woodblock            | 134     | Bird Tweet              |
| 101     | FX 5 (whistle)         | 109     | Koto              | 117     | Talk Box             | 135     | Telephone Ring          |
| 102     | FX 6 (bell)            | 110     | Bag Pipe          | 118     | Melodic Tom          | 136     | Halcyon                 |
| 103     | FX 7 (bells)           | 111     | Fiddle            | 119     | Synth Drum           | 137     | Applause                |
| 104     | FX 8 (cast-)           | 112     | Shabal            | 120     | Reverse Cymbal       | 138     | Gunsbat                 |

## General MIDI: Drums

MIDI "Drumkit" :=

- Sammlung von Drum/Percussion-Klängen
- je ein Klang pro Ton, z.B.

| Key# | Drum Sound         | Key# | Drum Sound     | Key# | Drum Sound    | Key# | Drum Sound     |
|------|--------------------|------|----------------|------|---------------|------|----------------|
| 35   | Acoustic Bass Drum | 47   | Low-Mid Tom    | 59   | Ride Cymbal 2 | 71   | Short Whistle  |
| 36   | Bass Drum 1        | 48   | Hi-Mid Tom     | 60   | Hi Bongo      | 72   | Long Whistle   |
| 37   | Side Stick         | 49   | Crash Cymbal 1 | 61   | Low Bongo     | 73   | Short Guiro    |
| 38   | Acoustic Snare     | 50   | High Tom       | 62   | Mute Hi-Conga | 74   | Long Guiro     |
| 39   | Hand Clap          | 51   | Ride Cymbal 1  | 63   | Open Hi-Conga | 75   | Claves         |
| 40   | Electric Snare     | 52   | Chinese Cymbal | 64   | Low-Conga     | 76   | Hi Wood Block  |
| 41   | Low Floor Tom      | 53   | Ride Bell      | 65   | High Timbale  | 77   | Low Wood Block |
| 42   | Closed Hi-Hat      | 54   | Tambourine     | 66   | Low Timbale   | 78   | Mute Cuica     |
| 43   | High Floor Tom     | 55   | Splash Cymbal  | 67   | High Agogo    | 79   | Open Cuica     |
| 44   | Pedal Hi-Hat       | 56   | Cowbell        | 68   | Low Agogo     | 80   | Mute Triangle  |
| 45   | Low Tom            | 57   | Crash Cymbal 2 | 69   | Cabasa        | 81   | Open Triangle  |
| 46   | Open Hi-Hat        | 58   | Vibraslap      | 70   | Maracas       |      |                |

- Auswahl: Rock/Pop und einige Klassik-Klänge
- unzureichend für aktuelle Musikformen

## General MIDI: Fazit

- grundlegendes Konzept hat sich bewährt
  - aber zu geringe Auswahl der Instrumente / Klänge
  - stark unterschiedliche Qualität der Geräte und Dateien
- => diverse Erweiterungen:
- Roland GS  
Instrumenten-Bänke (128\*128), Effekte
  - Yamaha XG  
Instrumenten-Bänke (128\*128\*128), Effekte, s.u.
  - General MIDI 2  
2 Drumkits, mehr Instrumente und Effekte
  - DLS  
Downloadable Sounds (vgl. SoundFont)

## XG: XG-plugin



S-YXG50: standalone Software-Synthesizer

- Funktionsumfang ähnlich dem XG50DB Hardwaremodul
  - 32 Stimmen, 48 KHz, 16 bit, 2 MB ROM, 676 Klänge, Effekte, ...
  - Sordius-XG: zusätzlich mit VL physical modeling synth
  - Latenz zu hoch zum Live-Spielen
- Demo per Download: [www.yamaha.co.uk/xg/html/midplug](http://www.yamaha.co.uk/xg/html/midplug)

- entsprechend: Roland SoundCanvas für GM/GS
- VST-Plugins: s.u.

Digitale Audioverarbeitung | WS 2000 | 18.205

## XG: Yamaha MU100R

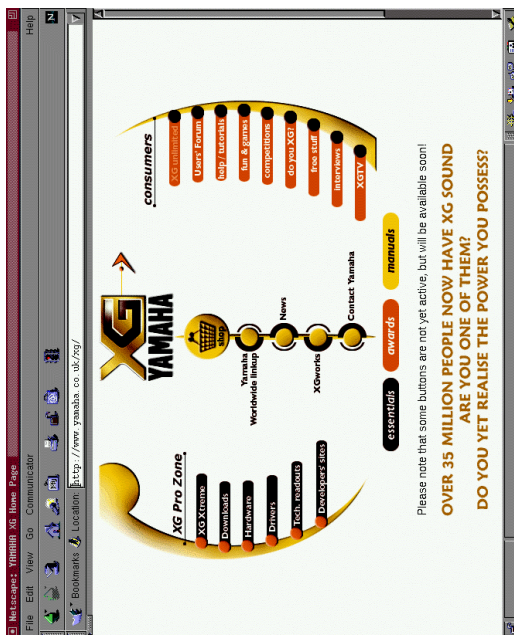


Racksynthesizer, 1998:

- 64-stimmig, 32-fach multitimbral, 32 MIDI-Kanäle
- 20 MByte Sample-ROM
- ca. 1500 "Instrumente", und 40 Drumsets
- GM, XG, Performance-Mode
- 7 separate Digitaleffekte, jeweils > 40 Algorithmen
- flexibles Routing der Instrumente zu Effekten
- VL physical modeling plugin
- VH vocal harmonizer plugin

Digitale Audioverarbeitung | WS 2000 | 18.205

## XG



Digitale Audioverarbeitung | WS 2000 | 18.205

## XG: Eigenschaften

XG := "eXtended General MIDI" := Erweiterung des GM-Standards

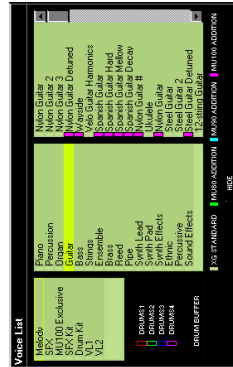
- mindestens 32-stimmig
- zusätzliche Instrumente: "Bänke"
- abwärtskompatibel / fall-back

zusätzliche Drumsets:

- z.B. "Analog", "Latin", "Jungle", ...
- Drumsets auf beliebigen MIDI-Kanälen

flexiblere Effekte:

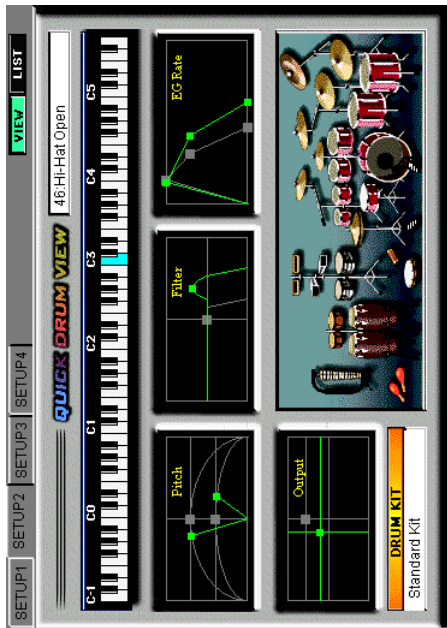
- Hall und Chorus mit vielen, feinen Varianten
- zusätzliche Variation-Effekte mit flexiblem Routing
- Eingänge für externe Signale



Digitale Audioverarbeitung | WS 2000 | 18.205



### XG: Drums

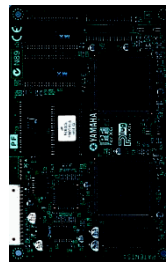


jeder einzelne Klang modifizierbar ...

### XG: Plugins

Yamaha "modular synthesis plugins"

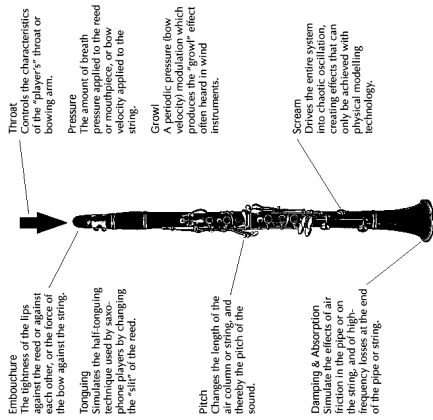
- Erweiterungskarten für XG-Geräte
- mit eigener Klangerzeugung



- VL Physical-Modeling Synthese, insb. Blasinstrumente
- VH "Vocal Harmonizer"
- PF samplebasierter Pianoexpander
- AN Virtuell-analoger Synthesizer
- DX Frequenzmodulation, kompatibel mit DX7

- flexible Erweiterung um andere Syntheseformen
- bisher keine ROM-Erweiterungen / Samplekarten

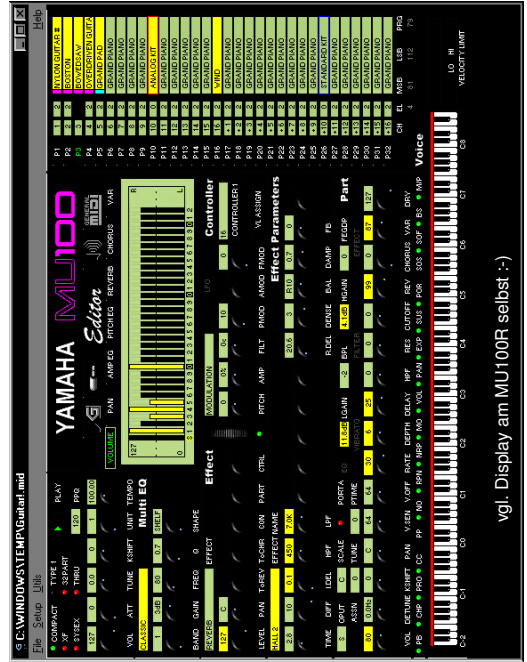
### XG: VL-Parameter



- Embouchure**  
Controls the tightness of the lips against the reed or against the bow against the string.
- Throat**  
Controls the characteristics of the "player's" throat or blowing air.
- Pressure**  
The amount of breath pressure applied to the reed or modulators or bow velocity applied to the string.
- Growl**  
A periodic pressure (bow velocity) modulation which produces a growl effect often heard in wind instruments.
- Pitch**  
Changes the length of the air column or string, and thereby the pitch of the sound.
- Damping & Absorption**  
Simulate the effects of air friction in the pipe or on the reed, and the frequency losses at the end of the pipe or string.

- "Blasinstrumente" mit realistischen Modulationsmöglichkeiten
- erfordert Breath-Controller

### XG: XGedit



vgl. Display am MU100R selbst :-)

## MIDI: Sequencer

- := Editor für MIDI-Daten
- := digitales Mehrspur-Tonbandgerät
- Aufnehmen einzelner MIDI-Spuren (overdub)
- Bearbeiten von MIDI-Daten, z.B. Quantisierung
- MIDI-Mischpult
- Verwalten der angeschlossenen externen MIDI-Geräte
- Cubase VST [www.steinberg.de](http://www.steinberg.de)
- Emagic Logic [www.emagic.de](http://www.emagic.de)
- Cakewalk [www.cakewalk.com](http://www.cakewalk.com)
- Pro Tools [www.digidesign.com](http://www.digidesign.com)
- und viele weitere ...

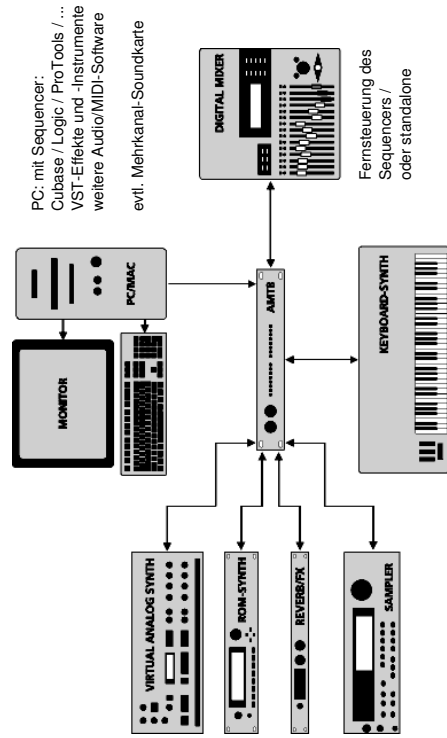
## Sequencer: Audio/MIDI-Integration

- reine MIDI-Aufnahmen "wenig lebendig"
  - Ergänzung um Solo-Instrumente und -Stimmen
  - separate Verwaltung der Audio- und MIDI-Daten mühsam
- => Integration von Audio- und MIDI-Daten
- leistungsfähige PCs ermöglichen ca. 10 .. 40 Audiospuren
  - bei samplegenauem Timing
  - diverse Audiofunktionen, z.B. Sampling, Pitch-Shifting, ...
  - integrierte Effekte: virtuelles Studio
- => "Harddisk-Recording"
- kein externes Bandgerät mehr nötig
  - Direktzugriff auf einzelne Spuren und Zeitpunkte

## virtuelles Studio: Konzept

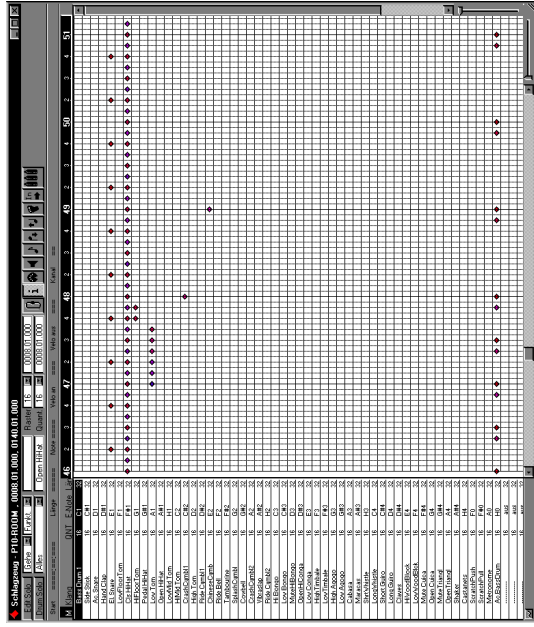
- |                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>virtuelles Studio</li> <li>• Mikrophone</li> <li>• AD-Wandler mit Verstärker</li> <li>• SW-Sampler/Synthesizer</li> <li>• SW-Mischpult</li> <li>• SW-"Plugins"</li> <li>• Festplatten (HD-Recording)</li> <li>• Einstellungen:</li> <li>• SW-Patchfelder / MIDI reproduzierbar (total recall)</li> <li>• speicherbar und automatisierbar</li> <li>• Cubase / Logic / ProTools / ...</li> </ul> | <p>herkömmliches Studio</p> <ul style="list-style-type: none"> <li>Mikrophone</li> <li>Vorverstärker</li> <li>Sampler/Synthesizer</li> <li>Mischpult</li> <li>Effektgeräte</li> <li>Tonbandmaschine</li> <li>Patchkabel / via MIDI</li> <li>nicht reproduzierbar</li> <li>nicht speicherbar</li> </ul> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## VST: Studio Setup



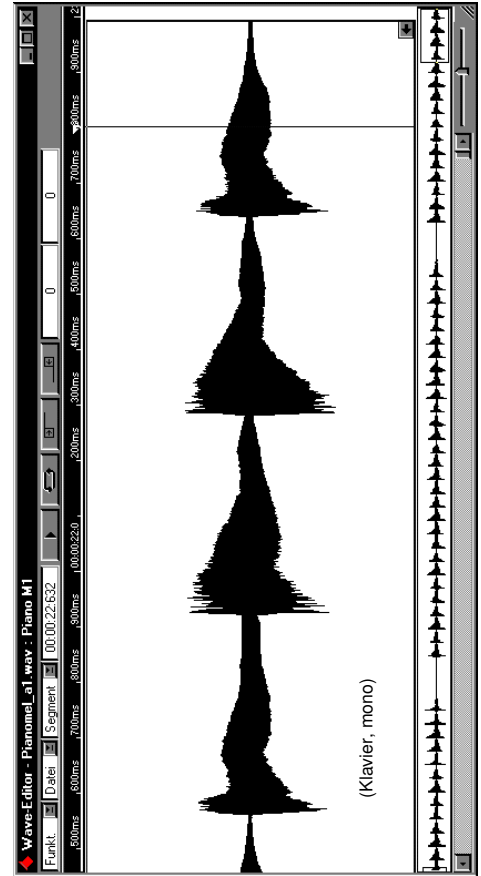
- bel. Kombinationen von externer HW und SW "plugins"

VST: Drum-Editor



Digitale Audioverarbeitung | WS 2000 | 18.205

VST: Cubase Wave-Editor

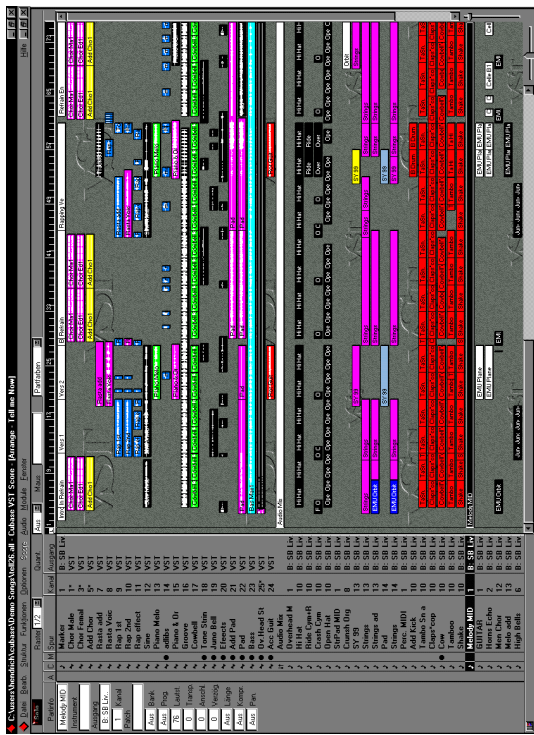


(Klavier, mono)

- alternativ externe Audio-Editoren (z.B. Wavelab)

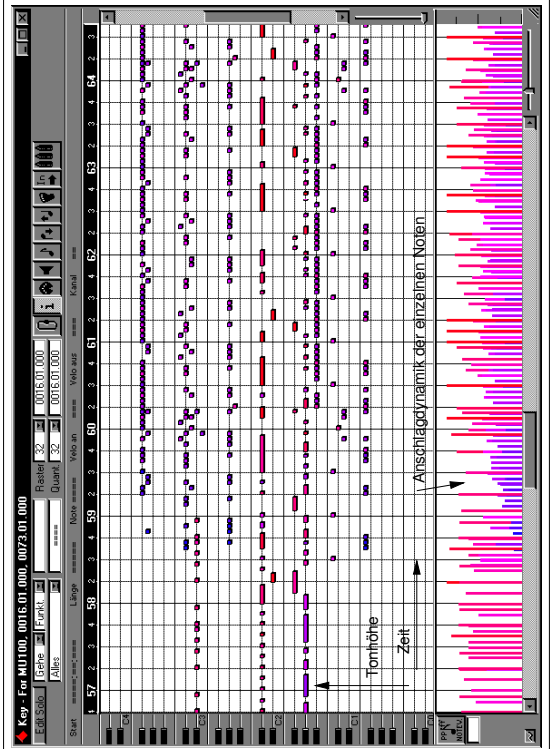
Digitale Audioverarbeitung | WS 2000 | 18.205

VST: Cubase Hauptfenster



Digitale Audioverarbeitung | WS 2000 | 18.205

VST: Cubase "piano-roll"

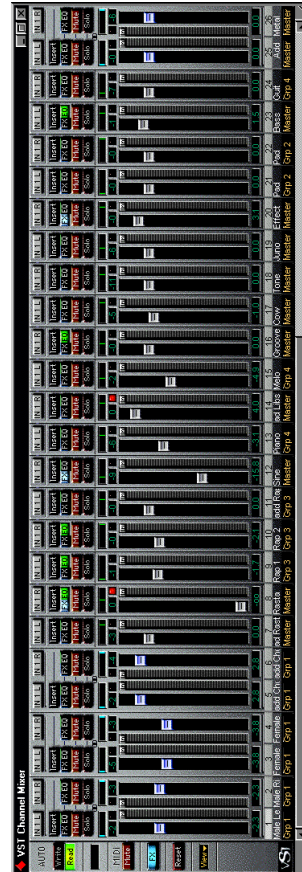


Digitale Audioverarbeitung | WS 2000 | 18.205

VST: Cubase Datenfluß

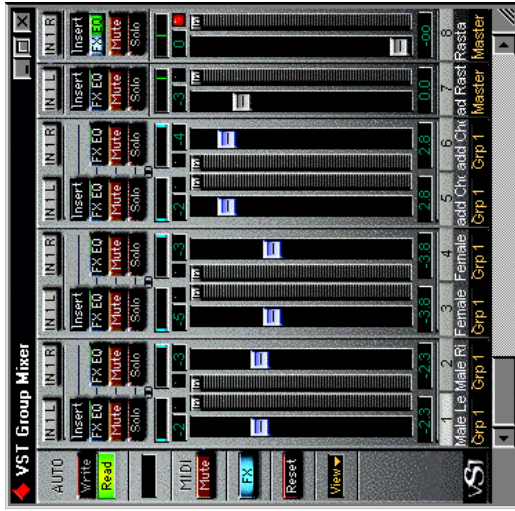
- Folie noch nicht fertig, grob:
- Audiospuren, externe MIDI-Instrumente, VST-Instrumente
  - Mischer, Gruppen-Mischer, Effekte, Dynamik, Master

VST: Cubase Mischer



- unbegrenzte Anzahl der Stereo-/Mono-Kanäle
- jeweils mit eigenen Effekten und Dynamikbearbeitung
- flexibles Gruppen- und Master-Routing
- aktuelle 1GHz-PCs erlauben ca. 40 Kanäle bei 32bit/48KHz

VST: Gruppen-Mischer

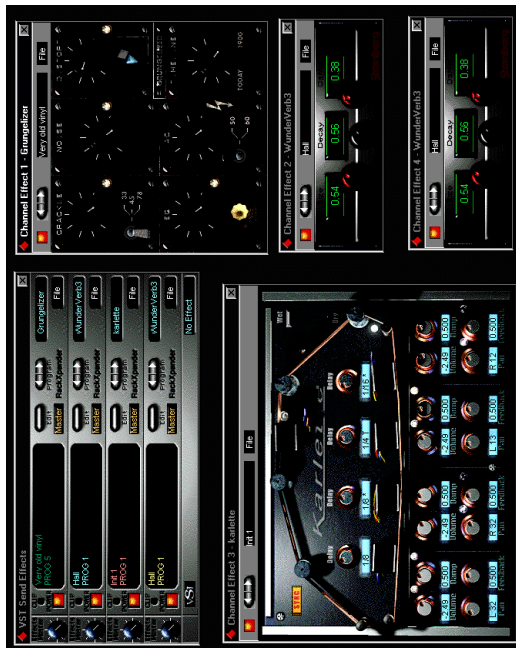


VST: Gruppen-Mischer, Dynamik





### VST: Effekt-Rack



### virtuelles Studio: Steinberg VST-2

Steinberg VST2: [\[www.steinberg.de\]](http://www.steinberg.de)

- Schnittstelle zwischen virtuellen Geräten
- Audio- und MIDI-Funktionen, samplegenau
- Integration in Cubase / andere Hostapplikationen
- realisiert als C++ Basisklasse, implementiert für PC/Mac/SGI
- 32-bit Gleitkomma für alle Datenoperationen
- ISSE-Optimierung
- zusätzliche GUI-Wrapper für Oberfläche der Plugins
- minimaler Overhead, optimale Performance
- als Standard etabliert

siehe Beispiel:

### VST: Instrument-Rack



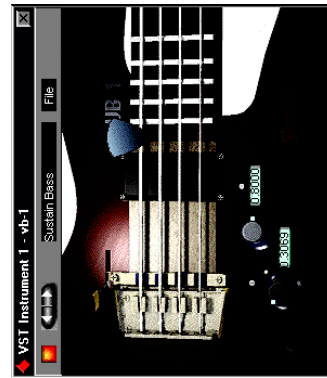
Neon: einfacher subtraktiver Synth

- zwei Instrumente mitgeliefert: Neon, VB1
- beliebig viele weitere Instrumente integrierbar
- verfügbar: z.B. LM4, Model\*E, Pro 52, B4, PPG, ...

### virtuelles Studio: VST-2 Plugin

```
#include "AGain.hh"
AGain::AGain(audioMasterCallback audioMaster)
 : AudioEffectX(audioMaster, 1, 1) // 1 program, 1 parameter
{
 fGain = 1.0; // default gain 0 dB
 setNumInputs(2); // stereo in
 setNumOutputs(2); // stereo out
 setUniqueId("AGain"); // unique name for this plugin
 canMono(0); // ok to feed with input with same values
 canProcessReplacing(); // accumulate / overwrite
 strcpy(programName, "default");
}
void AGain::setParameter(long index, float value)
{
 fGain = value;
}
...
void AGain::process(float **inputs, float **outputs, long n_samples)
{
 float *in1 = inputs[0]; float *out1 = outputs[0];
 float *in2 = inputs[1]; float *out2 = outputs[1];
 while(--n_samples >= 0) {
 (*out1++) += (*in1++) * fGain; // accumulating: Mixer
 (*out2++) += (*in2++) * fGain; // should use ISSE/3dnow
 }
}
...

```



VB-1: physical modeling bass

## Digitale Wasserzeichen

Verschlüsselung vs. Wasserzeichen

- Konzept
- Angriffe
- Beispiel Memorystick

Markierung von Audiodateien:

- EQ, Filter
- LSB-Verfahren
- Korrelations-Verfahren, Echo-Verfahren
- SDMI
- HackSDMI-Wettbewerb

Digitale Audioverarbeitung | WS 2000 | 18.205

## Literatur

- Proceedings of the IEEE, special issue on "identification and protection of multimedia information", 07/1999
- IEEE Trans. Signal Processing, spec. issue, "digital watermarking", 09/2000
- "Information hiding", Lecture notes in computer science, LNCS 1174, Cambridge 1996, K-INF-23262
- Cambridge security group, [www.cl.cam.ac.uk](http://www.cl.cam.ac.uk)
- J. Boeuf, J.P. Stern: An analysis of one of the SDMI candidates [www.julienstern.org](http://www.julienstern.org)
- [www.research.ibm.com/journal/sj/mit/sectiona/bender.html](http://www.research.ibm.com/journal/sj/mit/sectiona/bender.html)
- diverse Firmen und Organisationen, u.a.:  
[www.sdmi.org](http://www.sdmi.org)      [www.hacksdmi.org](http://www.hacksdmi.org)  
[www.musiccode.com](http://www.musiccode.com)      [www.watermarkingworld.org](http://www.watermarkingworld.org)  
[www.audiotrack.com](http://www.audiotrack.com)      [www.4centify.com](http://www.4centify.com)  
[ftp://ftp.cryptocd.com/pub/cryptocd/](http://ftp.cryptocd.com/pub/cryptocd/)

Digitale Audioverarbeitung | WS 2000 | 18.205

## Das Problem

- **Riesenmarkt:** USA sales 1997: CDs 9.915 M\$, CCs 1.523 M\$ (IBM)

Analogtechnik:

- Masterbänder altern
- jede Kopie schlechter als die Vorlage
- schlechte Qualität bei Consumertechniken (VHS, CC, ...)

Digitaltechnik:

- alle Kopien identisch mit Vorlage
- Alterung durch Kopieren kompensierbar
- bisheriger Kopierschutz sinnlos (S/PDIF copy-bit)
- auch billige Geräte/Recorder bieten 1:1 Kopien
- extrem gutes Preis/Leistungsverhältnis

=> (Raub-) Kopieren" nimmt zu (Napster, DivX, ...)



Digitale Audioverarbeitung | WS 2000 | 18.205

## Die Hoffnung

- Riesenmarkt "Musik"
- digitale Daten perfekt kopierbar

Verfahren zum Kopierschutz notwendig:

- 1) Verschlüsselung der Daten  
Zugriffskontrolle: Abspielen, Exportieren, ...
  - 2) Markierung von Daten mit Urheber-Informationen  
Erkennung von Raubkopien
  - 3) Personalisierung der Daten  
Zurückverfolgen von Raubkopien
- einige Verfahren bereits am Markt:  
LiquidAudio, WindowsMedia, SDMI, ...

Digitale Audioverarbeitung | WS 2000 | 18.205

## Begriffe

### Kryptographie

- Nachricht komplett verschlüsseln
- sichere Verfahren bekannt

### Steganographie

- geheime Nachricht in offener Nachricht verbergen
- Morsecode mit {i,j} {f,t}
- Formatierung, z.B. Binärcode mit space, tab

### Wasserzeichen ("watermark")

- Sichern des Urheberrechts, Angaben des Käufers, etc.
- offen oder versteckt

## Verschlüsselung

- symmetrisch: DES, IDEA, ...
- asymmetrisch (public key): RSA, ...
- viele Verfahren gelten als sicher
- abhängig von Schlüssellänge und "qualität"
- obwohl die Algorithmen bekannt sind

### für Audiodateien:

- gängige Algorithmen eignen sich auch für Audio
- zunehmend verwendet, z.B. in WindowsMedia, LiquidAudio, ...
- auch in Hardware: Sony MagicGate MemoryStick
- aber: einmal entschlüsselte Daten können (raub)kopiert werden

## MemoryStick

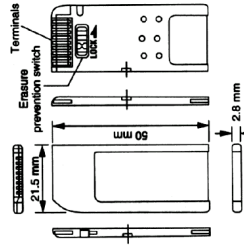


Figure 2. Memory Stick dimensions (mm).

Table 1. Memory Stick specifications.

| Item                  | Description                                    |
|-----------------------|------------------------------------------------|
| Capacity (Mbytes)     | 4, 8, 16, 32, and 64 (currently); 128 (future) |
| No. of connector pins | 10                                             |
| Interface type        | Serial                                         |
| Serial clock          | 20 MHz (maximum)                               |
| Write speed           | 1.5 Mbytes/s (maximum)                         |
| Read speed            | 2.45 Mbytes/s (maximum)                        |
| Power source voltage  | 2.7 V to 3.6 V                                 |
| Operating current     | 45 mA (average); 130 $\mu$ A (standby)         |
| Dimensions            | 21.5 mm wide x 50 mm long x 2.8 mm thick       |
| Weight                | ~4 grams                                       |

### Sony memorystick (1998):

- Flash-RAM basiertes Speichermedium
- kompakte Abmessungen, robustes Gehäuse
- als Konkurrenz zu SMC/MMC Speicherkarten
- "MagicGate"-Erweiterung: mit on-chip Verschlüsselung [IEEE Micro 7/8-2000, 40]



## MemoryStick: Konzept

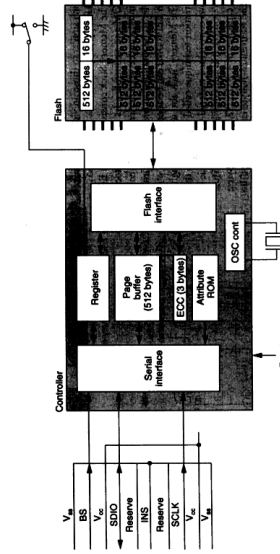


Figure 3. System configuration.

- für Consumergeräte: klein, robust, wenig Pins
- einheitliches Dateisystem, basiert auf FAT-16
- eingebauter Controller zur Ansteuerung des FLASH-RAM
- für Multimedia: Blockgröße 8 .. 16 KB

### MemoryStick: serielles Protokoll

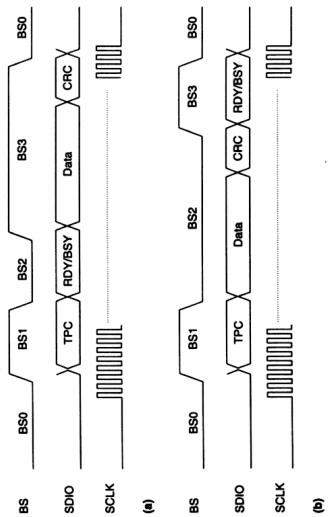


Figure 4. Timing example: read protocol (a) and write protocol (b).

- serielle Datenübertragung, SCLK bis 20 MHz
- nur drei Leitungen BS, SDIO, SCLK
- Lesen bis 1.5 MB/s, Schreiben bis 2.45 MB/s

### MemoryStick: Anwendungen

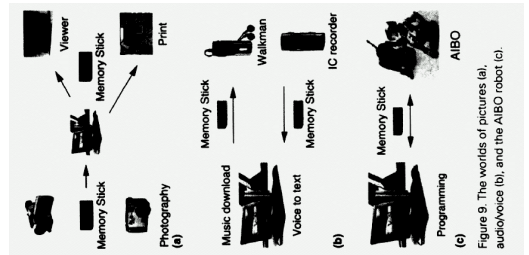


Figure 9. The worlds of pictures (a), audio/voice (b), and the AIBO robot (c).

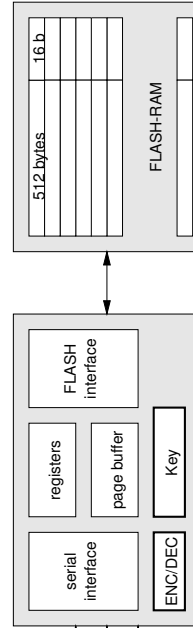
### MemoryStick: Vergleich

bisher jeder Hersteller mit eigenem Format:



- Memorystick
  - Smart Media Cards
  - Compact Flash Cards
  - SIM-Cards
  - PCMCIA usw.
- jedes Format hat Vor- und Nachteile
  - z.B. Frage controller / raw memory
  - zukünftige Marktentwicklung unklar
  - Übersicht: siehe Digitalkamera-Test in ct 11/2000

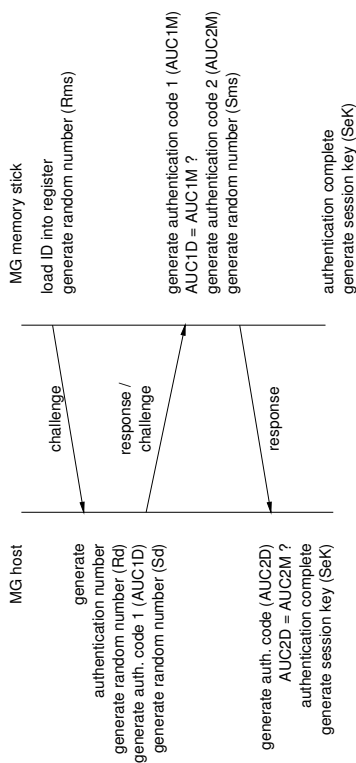
### MemoryStick: MagicGate



MagicGate := erweiterter MemoryStick mit Verschlüsselung, Ende 1999

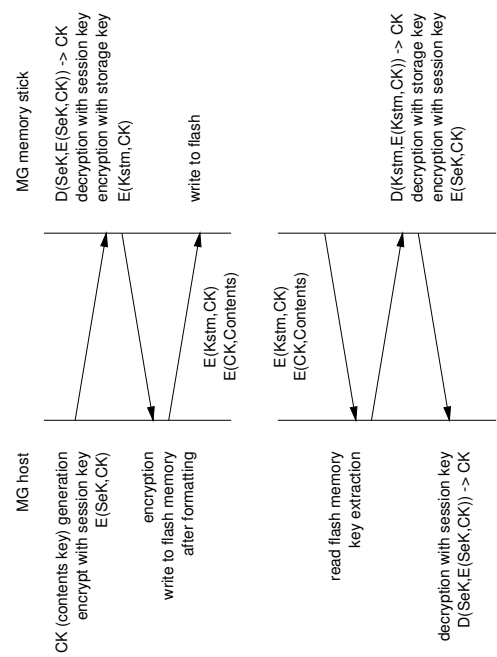
- eindeutige Seriennummer in jedem MG-MemoryStick
- erlaubt Identifikation des Mediums und der Daten
- Host übernimmt die Ver-/Entschlüsselung
- geringer Hardwareaufwand im Memorystick-Controller
- dadurch geringe Kosten

## MemoryStick: Authentifizierung



- basiert auf der (eindeutigen) ID des MG-Memorystick
- erzeugt "session key" für die Ver-/Entschlüsselung

## MG-MemoryStick: Read/Write

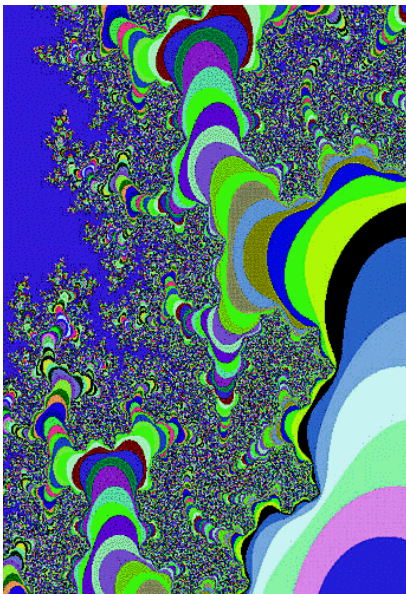


## Steganographie

"information hiding" :

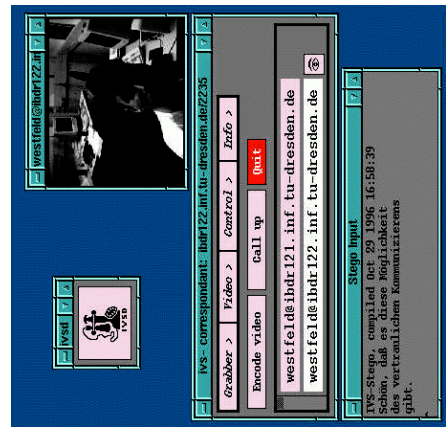
- geheime Nachricht in offener Nachricht verbergen
- bei Bedarf: geheime/offene Nachricht zusätzlich verschlüsseln
- z.B. Umgehen von Kryptographie-Exportverboten
- diverse Algorithmen und Tools erhältlich
- aber: keine sicheren Verfahren bekannt
- nur sicher, solange:
  - keine geheime Nachricht vermutet wird
  - der eingesetzte Algorithmus geheim bleibt
- viel schwieriger als Kryptographie:
  - weil das Original "unverändert" aussehen soll

### Steganographie: Mandelsteg



- Verstecken von Daten in veränderten (Mandelbrot-) Fraktalen  
<http://ftp.cslua.berkeley.edu/pub/cypherpunks/steganography/>

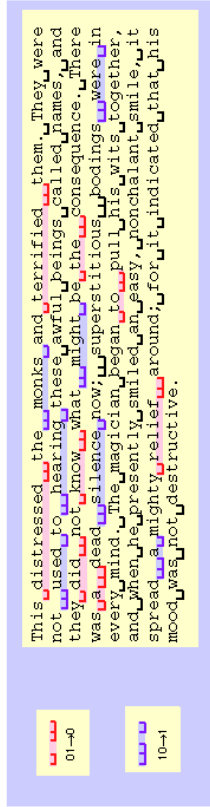
### Steganographie: Videokonferenz



- Übertragen von "Nebenabreden" ...  
[www.inf.tu-dresden.de/~hfd/pub/1997/FFWW\\_97ITSStego.pdf](http://www.inf.tu-dresden.de/~hfd/pub/1997/FFWW_97ITSStego.pdf)

### Steganographie: Textdateien

Figure 30 Data hidden through justification (text from *A Connecticut Yankee in King Arthur's Court* by Mark Twain)



[Bender 98]

Textdateien sind besonders schwer zu sichern:

- hinzugefügte Zeichen zerstören die Nutzinformation
- Wasserzeichen nur über Formatierung
- z.B. Tab/Space-Verfahren

### Steganographie: Textdateien

beim Papierausdruck mehr Möglichkeiten:

- Zeilenabstände oder Wortabstände modulieren (z.B. ~1/300)
- modifizierte Fonts
- fällt normalerweise nicht auf
- übersteht Vergrößerung, mehrfaches Kopieren
- übersteht Drucken, Scannen, OCR
- aber ASCII-Export zerstört die Info

## Wasserzeichen

Wasserzeichen:

- Monogramm/Logo auf/in jedem Blatt
- seit etwa 1500
- als Qualitätsnachweis des Papiers
- oder zur Authentifizierung (Banknoten, Ausweis, Fahrkarten, etc)
- Fälschung stark erschwert
- Entfernung praktisch unmöglich

digitale Wasserzeichen:

- sichtbare oder unsichtbare Markierung von Daten
- bisherige Verfahren noch wenig robust

Digitale Audioverarbeitung | WS 2000 | 18.205

## Wasserzeichen: Papier (um 1550)



Fig. 6. Monograms figuring TGE RG (Thomas Goodrich Eliensis Bishop of Ely, England – and Remy/Renniguis Guedon, the paper-maker). One of the oldest watermarks found in the Cambridge area (c.1550). At that time, watermarks were mainly used to identify the mill producing the paper; a means of guaranteeing quality. Courtesy of Dr E. Leedham-Green, Cambridge University Archives. Reproduction technique: beta radiography.

Digitale Audioverarbeitung | WS 2000 | 18.205

## Wasserzeichen: Sony SACD

**Format Security**

Pirated copies of discs threaten music companies, producers and musicians alike. In addition, consumers need protection from counterfeit discs. That's why authorized SACDs are identified by both visible and invisible watermarks. The visible watermark is a faint image on the signal side of the disc, made by a laser engraving process. Invisible watermarks warn consumers of unauthorized discs. Invisible watermarks authenticate the discs prior to playback. If the SACD player cannot read the watermark, the SACD will not play.

The logos on this disc are not just decorative. They are digital watermarks -- valuable tools in copyright protection.

Sony Electronic's Super Audio Compact Discs are a registered trademark of Sony Electronics Inc. All Rights Reserved. [Learn More About Us](#)

Digitale Audioverarbeitung | WS 2000 | 18.205

- Verfahren nicht dokumentiert

## Anforderungen an Wasserzeichen

Wasserzeichen:

- ausreichende Datenrate für Kennzeichnung
- copy prohibit: 1 bit, ISBN: 10 Dezimalstellen
- Personalisierung: > 128 bit notwendig
- keine oder geringe Beeinträchtigung des Nutz- (Audio-) Signals

Robustheit:

- gegen elementare Signalverarbeitung
- gegen psychoakustische Signalverarbeitung
- gegen möglichst viele "Angriffe"
- Entfernung nur bei gleichzeitiger Verschlechterung des Nutzsignals
- gegen Fälschung

Digitale Audioverarbeitung | WS 2000 | 18.205

## Wasserzeichen: Audio

Möglichkeiten bei Audiodateien:

- typ. Datenrate 64 kbps (MP3) bis 1.5 Mbps (CDDA)
- Wasserzeichen: 100+ bits, ca. alle 10 Sekunden wiederholen
- "externe" Markierung (Chunk-Dateiformate)
- Filterung der Daten (Notch-Filter)
- LSB-Techniken
- Phasenverschiebungen
- Frequenzverschiebungen
- Spread-Spectrum
- Echo-Markierung
- Kombinationen dieser Verfahren

## typische Angriffe

- einfache Angriffe:
- einfache digitale Filter
  - Mischen mehrerer Signale oder mit Rauschen
  - Lautstärkeänderung, Dynamikänderung
  - sample-rate conversion
  - A/D-D/A Konvertierung
  - Tempoänderung, pitch-shifting
  - MP3-Kodierung usw.
  - Kombination mehrerer Verfahren (vgl. StirnMARK)

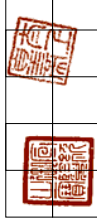
gezielte Angriffe möglich, sobald Algorithmus bekannt:

- Angreifer kann sehr viel Rechenzeit investieren
- praktisch nicht zu verhindern

## Mosaic-Attack

"Mosaic-Attack":

- Wasserzeichen erfordert Mindestlänge der Nutzdaten
- lange Dateien in viele kurze zerlegen
- Wasserzeichen wird verstümmelt
- für Bilddaten bereits Tools verfügbar
- Verfahren eignet sich auch für Audio
- z.B. als Verfahren gegen Web-Robots auf der Suche nach geklauten Abbildungen



## Interpretation-Attack

"Interpretation-Attack":

- die meisten Algorithmen sind "additiv"
- auch verschiedene Verfahren kombinierbar
- Angreifer fügt eigenes Wasserzeichen hinzu

Original:  $d$   
 Original + Wasserzeichen:  $d + w$   
 Pirat verbreitet:  $d + w + x$   
 Pirat behauptet:  $d + w$  ist das Original

- => Priorität der Urheberschaft?!
- => Reihenfolge der Wasserzeichen beweisbar?!



### Collusion-Attack

Überlagerung:

- Angriff gegen "personalisierte" Dateien
- Sammeln von vielen Varianten für eine Datei
- Mittelung all dieser Dateien

Original:

personalisierte Dateien:  $d_1 = d + w_1, d_2 = d + w_2, \dots, d_n = d + w_n$

Mittelung:  $D = 1/n * (d + d_1 + \dots + d + w_1 + w_2 + \dots + w_n)$

Nutzdaten bleiben erhalten, Wasserzeichen "mitten sich raus"

=> Nachweis aller einzelnen Wasserzeichen ?!

=> Robustheit und Skalierung für großes n ?

### Radio: mit Kopierschutz und Marken . . .

auch analoge Medien sind geschützt:

Radiosender jederzeit identifizierbar:

- Jingles
- besondere EQ-Einstellungen
- extreme Dynamikkompression

trotz zweifelhafter Qualität:

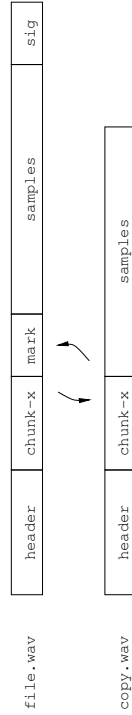
Mitschneiden unmöglich:

- Moderator spricht in Anfang und Ende jedes Songs
- Titel werden nicht ausgespielt
- Titel werden überblendet
- usw.

### externe Marken

Wasserzeichen im Header/in Kommentaren:

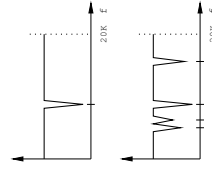
- einfachste Realisierung, etwa zusätzlicher Chunk im WAV-Format
- Nutzdaten werden nicht gestört
- trivial entferbar (s. S/PDIF Copy-Bit) und fälschbar
- aber Kombination mit digitaler Signatur möglich



### Notch-Filter

Signal mit Kerbfiltern (notchfilter) bearbeiten

- sehr schmale Notchfilter (z.B. 1/100 Oktave)
- kaum hörbar
- aber mit FFT sofort erkennbar
- evtl. mehrere Bänder sperren



- skaliert nicht auf hohe Anzahl verschiedener Wasserzeichen
- leicht entferbar
- sehr leicht fälschbar

## LSB-Techniken

Wasserzeichen im LSB der Nutzdaten kodieren:

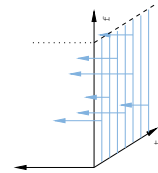
```
sample[t] = sample[t] & 0xfffff
 + mark[t] & 0x0001;
```

- sehr einfach zu realisieren
- fällt in (Pop-) Musik nicht auf
- in leisen Passagen evtl. hörbar
- sehr hohe Bitrate für das Wasserzeichen möglich
- Konflikt mit Dithering-Verfahren
- anfällig gegen Rauschen (z.B. durch DA-AD Wandlung)
- sehr leicht entfernbar, evtl. fälschbar
- bei 24-bit (DVD-Audio) auch mehrere Bits nutzbar

## Spread-Spectrum

schmalbandiges Signal in breitbandigem Signal verstecken

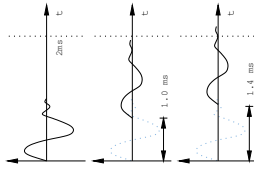
- Nutzsignal wird auf mehrere Frequenzbänder aufmoduliert
- Frequenzbänder werden ständig gewechselt
- Auswahl der Frequenzbänder pseudozufällig
- Sender und Dekoder verwenden gleiche Zufallszahlen
- seit WW2 militärisch genutzt
- GSM, DECT Mobiltelefone, GPS, usw.
- ohne Kenntnis der Zufallszahlen nicht detektierbar
- Nutzsignal detektierbar, auch wenn  $\ll$  Rauschen
- unempfindlich gegen einfache Angriffe
- sehr empfindlich gegen Timing-Veränderungen



## Echo-Marking

Wasserzeichen als Echo im Signal verstecken:

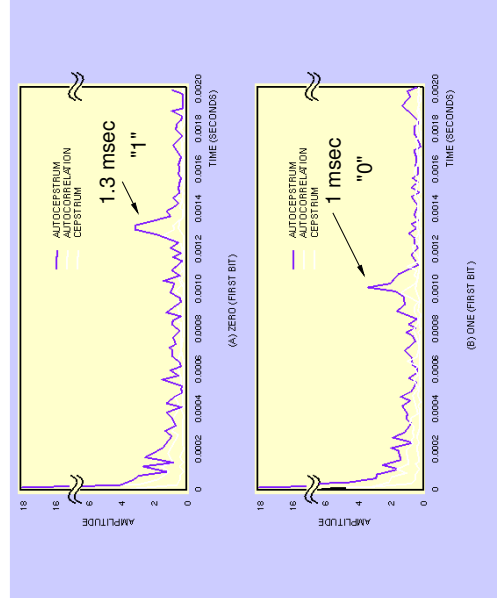
- kurze Echos sind kaum wahrnehmbar
- eignen sich damit als Kennzeichnung:
  - Bit 0: 1.0 msec Echo
  - Bit 1: 1.4 msec Echo
- Detektion erfordert Analyse der Echos
- sehr robust gegen alle einfachen Angriffe
- Entfernung des Echos sehr aufwendig
- aber machbar ...



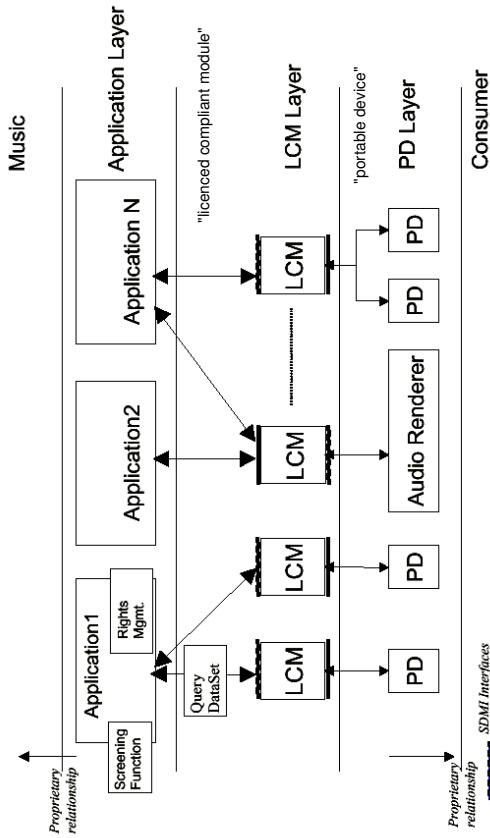
<http://www.research.ibm.com/journal/sj/mit/sectiona/bender.html>

## Echo-Marking

Figure 28 Result of autocorrelation and autocorrelation for (A) "zero" and (B) "one" bits

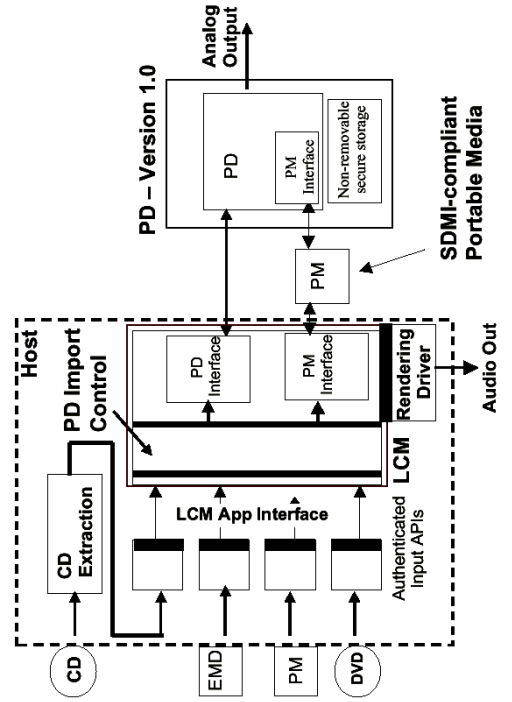


SDMI: layers



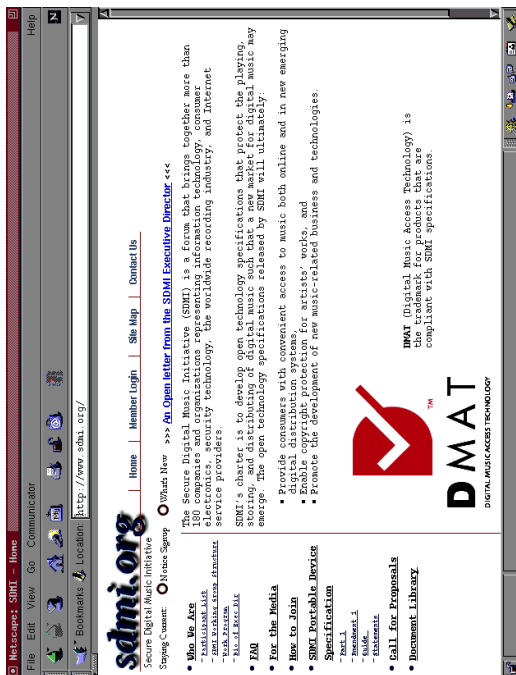
Digitale Audioverarbeitung | WS 2000 | 18.205

SDMI: devices



Digitale Audioverarbeitung | WS 2000 | 18.205

SDMI: homepage



Digitale Audioverarbeitung | WS 2000 | 18.205

SDMI: Konzept

SDMI := "Secure Digital Music Initiative"

- Kopierschutz für digitale Audiodaten:
  - gegen Raubkopien bzw. das Abspielen von Raubkopien
  - Entwicklung entsprechender Algorithmen und Geräte
  - Kooperation von ca. 200 Firmen
- Trennung von digitalem (sicheren) und analogem Bereich
- Erkennen von "compressed data" (d.h. insb. MP3)
- Umstieg "legacy" auf SDMI in mehreren Phasen
- Einsatz von Verschlüsselung
- Erkennung von Raubkopien über digitale Wasserzeichen
- Test der ersten Algorithmen Ende 2000

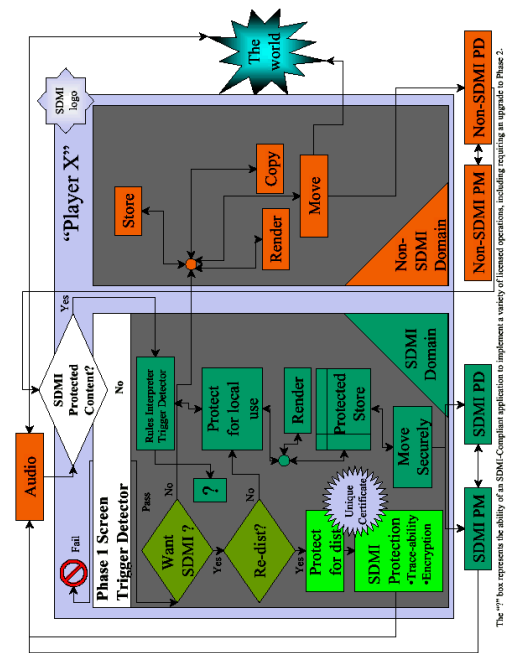
Digitale Audioverarbeitung | WS 2000 | 18.205

### SDMI: Matrix

| Capabilities                                                                           | Today (Non-SDMI Devices) | SDMI (Devices in Holiday 99) | SDMI (Future) |
|----------------------------------------------------------------------------------------|--------------------------|------------------------------|---------------|
| Download and Play current digital music tracks including MP3s                          | ✓                        | ✓                            | ✓             |
| Download and play SDMI digital music tracks                                            |                          | ✓                            | ✓+            |
| Transfer personal CD collection to a PC                                                | ✓                        | ✓                            | ✓             |
| Transfer current digital music tracks from PC to a portable device                     | ✓                        | ✓                            | ✓             |
| Transfer SDMI digital music tracks from PC to portable device                          |                          | ✓                            | ✓             |
| Share current digital music tracks                                                     | ✓                        | ✓                            | ✓             |
| Enable sharing of SDMI digital music tracks                                            |                          | ✓                            | ✓+            |
| Enable independent artists, church choirs, etc. to create and distribute digital music | ✓                        | ✓                            | ✓             |
| Explicitly supports copyright / rights management for digital music distribution       |                          | ✓                            | ✓+            |

- Abspielen und Kopieren von "legacy" Medien erlaubt
- aber "neue" Medien (ab. ca. 2000) geschützt

### SDMI: domains



### SDMI: Phasen

Phase 1:

- erste Gerätegeneration, mit Option zum Upgrade auf Phase 2
- unmarkierte Dateien können abgespielt werden
- markierte Dateien werden zurückgewiesen
- dann Upgrade auf Phase 2 notwendig

Phase 2:

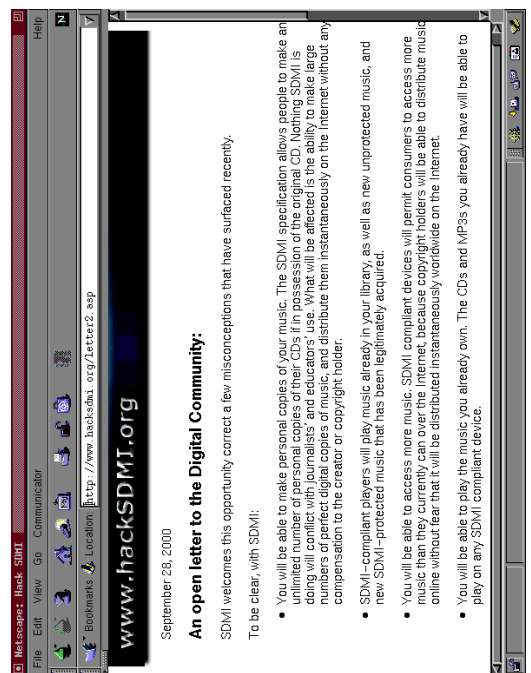
- spielt unmarkierte und "heile" markierte Dateien
- erkennt komprimierte markierte Dateien, spielt diese nicht
- bisher nicht voll spezifiziert

## SDMI: HackSDMI

Test der SDMI Algorithmen (Kandidaten) notwendig

- öffentlicher Wettbewerb, [www.hacksdmi.org](http://www.hacksdmi.org)
- Preisgeld von \$10.000 für das "Knacken" der Algorithmen
- für jedes der vorgestellten Verfahren je drei Wav-Dateien:
  - Song A Original, ohne Wasserzeichen
  - Song AW mit Wasserzeichen
  - Song BW mit Wasserzeichen
- und ein Orakel:
  - Upload "geknackter" Versionen B' von BW
  - Erkennung des Wasserzeichens
  - Bewertung der Audioqualität

## HackSDMI: homepage



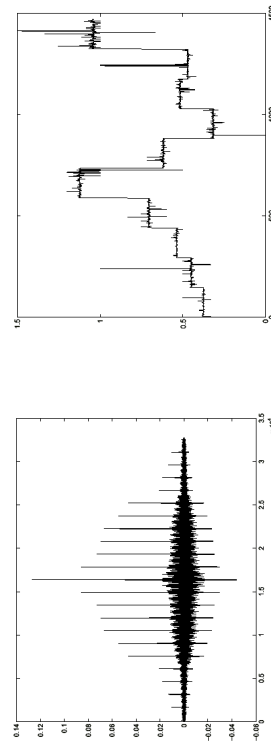
## HackSDMI: Analyse und Angriff

- gegeben: Demodateien A, AW, BW
- und: Wasserzeichen W ist für alle Dateien gleich (!)

Angriffsprinzip:

- Wasserzeichen extrahieren,  $W = (A - AW)$
- anschließend analysieren: Autokorrelation usw.
- Spezifikation fordert: Periode < 15 sec. anschließend:
- Prinzip des Algorithmus erkennen
- passenden Encoder/Decoder schreiben
- Wasserzeichen gezielt angreifen ("surgical attack")
- notfalls "random attack": bis Orakel(Player) die Datei akzeptieren

## HackSDMI: Autokorrelation



- links: Autokorrelation des Wasserzeichens (Differenz AW-A)
- offensichtliche Korrelation alle 1470 Samples, => Periode 1470
- rechts: Korrelation des um 1470 Samples verzögerten W-Signals
- alle 147 Samples mit veränderter Amplitude
- vermutlich W proportional zum Nutzsignal,  $\|W(i)\| \sim \|S(i)\|$

## HackSDMI: vermutete Algorithmen

**Algorithm 1** Marking algorithm: inputs:  $w \in [-1, 1]^{1470}$ ,  $s \in [-1, 1]^m$

```

Output and skip start samples from the original song
while The song is not over do
 $s \leftarrow$ the next 1470 samples of the song
 for $j = 1$ to 10 do
 $s[j] \leftarrow s[j] + \beta \|s[j]\| w[j]$
 end for
Output s
end while

```

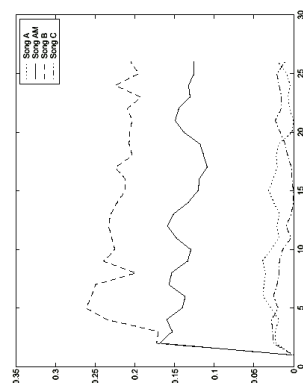
**Algorithm 2** Detection algorithm: inputs:  $w \in [-1, 1]^{1470}$ ,  $s^i \in [-1, 1]^m$ ,  $p$ ,  $\delta$

```

Skip start samples (possibly resynchronize by correlation)
while The song is not over do
 $sum \leftarrow 0$
 Get the next p chunks of 1470 samples
 for Each of these chunks do
 $g \leftarrow$ the next 1470 chunk
 for $j = 1$ to 10 do
 $s[j] \leftarrow s[j] / \beta \|s[j]\| w[j]$
 end for
 $sum \leftarrow sum + g$
 end for
 $Q = sum.w$
 if $Q > \delta$ then
 Outputs "mark found"
 end if
end while

```

## HackSDMI: Analyse des Angriffs



- Analyse der Signale A, AW, BW, und des rekonstruierten B
  - links: Ausgangssignal des Detektionsalgorithmus
  - rechts: Spektrum des Wasserzeichen-Signals
- => Wasserzeichen ohne Qualitätsverlust entfernt

## HackSDMI: Status

- hacksdmi.org Website derzeit nicht mehr erreichbar
- keine eigenen Experimente mehr möglich
- Preisgeld für zwei Angriffe ausgezahlt
- angeblich alle Verfahren "geknackt":  
<http://www.cs.princeton.edu/sp/sdmi/>  
<http://www.julienstern.org/sdmi/>
- derzeit keine "sicheren" Verfahren bekannt
- einige Sicherheit nur bei "geheimem" Algorithmus
- auch dann trügerisch (vgl. DeCSS)

## Feedback

- Hinweise / Ergänzungen / Berichtigungen
  - Wünsche / Anmerkungen
  - neue Themenvorschläge
  - hörenswerte neue CDs / DVDs / MP3s / MIDIs
  - interessante neue Software, neue Geräte
  - Interesse an Studien/Diplomarbeiten !?
  - Vorschläge für Projekte !? (auch für das Baccalaureat)
- => bitte an: [hendrich@informatik.uni-hamburg.de](mailto:hendrich@informatik.uni-hamburg.de)
- vielen Dank für das Interesse!