Der Einsatz komplexer Algorithmen zur Realisierung universeller Abtastempfänger in FPGA–Technik

Dem Fachbereich 19 der
Technischen Hochschule Darmstadt
zur Erlangung der Würde eines
Doktor–Ingenieurs (Dr.–Ing.)
vorgelegte Dissertation

Dipl.–Ing. Uwe Meyer–Bäse
geboren am 10.7.1964

Referent:
Korreferenten:

Tag der Einreichung:
Tag der mündlichen Prüfung:

D17
Darmstädter Dissertation

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Datentechnik der Technischen Hochschule Darmstadt.

Meinem Referenten, Herrn Prof. Dr.–Ing. W. Hilberg, möchte ich an dieser Stelle ganz besonders für seine Förderung und Betreuung meiner Arbeit danken. Als Erfinder der Funkuhrtechnik konnte Herr Prof. Dr.–Ing. W. Hilberg durch zahlreiche Diskussionen meine Forschungen wertvoll unterstützen.

Den Herren Prof. Dr.–Ing. A. Kessler und Prof. Dr.–Ing. H. Eveking gilt mein Dank für ihr Interesse an dieser Arbeit und die Übernahme des Korreferates. Insbesondere möchte ich mich bei Herrn Prof. Dr.–Ing. H. Eveking für die an seinem Institut der Universität Frankfurt durchgeführten Untersuchungen zu den Kapiteln 3 und 5 bedanken.

Den technischen Zeichnerinnen, Frau K. Schork–Jakobi und Frau I. Klenk, danke ich für die Erstellung zahlreicher Diagramme, durch die diese Arbeit sich wesentlich anschaulicher gestaltet.

Meinen Kollegen, insbesondere den Herren Dipl.–Ing. V. Nachtwey, Herrn Dipl.–Ing. J. Sendler, Herrn Dipl.–Ing. D. Schulz, Herrn Dipl.–Ing. S. Wolf und Dipl.–Ing. R. Watzel, sowie den ehemaligen Mitarbeitern des Institutes, Herrn Dipl.–Ing. S. Höreth und Herrn Dr.–Ing. J. Wietzke, danke ich für die gute Zusammenarbeit und die zahlreichen wertvollen Diskussionen und Anregungen.

Nicht zuletzt danke ich meinen Eltern für die Ermunterung zur Durchführung dieser Arbeit und meiner Ehefrau Dipl.–Ing. Anke Meyer–Bäse für erhebliche Unterstützung in jeglicher Beziehung.

Inhaltsverzeichnis

1 Grundlegende Betrachtungen **3**

1.1 Darstellung von Bandpaß–Signalen durch I– und Q–Phase 3

1.1.1 Klassische Erzeugung von I– und Q–Phase durch die Hilbert–Transformation 5

1.1.2 Kohärente Demodulation 6

1.1.3 Demodulation mit Hilfe von I– und Q–Phase 7

1.2 Beschreibung der LW–Datensignale 7

1.2.1 Der Zeitzeichensender DCF77 9

1.2.2 Die Wetterbildsender DCF37 und DCF54 10

1.2.3 Signale des Textsenders DCF54 12

1.3 Kriterien zur Beurteilung von PGAs 12

2 Methoden der Frequenzsynthese **17**

2.1 Klassische analoge Verfahren zur Frequenzsynthese 17

2.2 Direkte Digitale Synthesizer 18

2.2.1 Direkte Digitale Synthesizer mit Akkumulator 18

2.2.2 Der Frequenzteiler 23

2.2.3 Multiplikative Frequenzerzeugung 24

2.2.4 Abschließender Vergleich der DDS–Verfahren 25

2.3 Hybride Verfahren zur Frequenzsynthese 25

2.3.1 Die PLL zur Frequenzsynthese 26

2.3.2 Modifizierte PLL–Synthesizer 27

2.3.2.1 Akkumulator–Synthesizer als PLL–Referenz 28

2.3.2.2 Fraktionale M–Synthese 29

2.3.3 Neuer Synthesizer mit Akkumulator–Rest–Division 31

2.3.3.1 Verallgemeinerung 31

2.3.3.2 Hybride Struktur zur Phasenfehlerkompensation 33

2.3.3.3 Hardware–Konzepte 33

2.3.3.4 Beschreibung des Testaufbaus 36

2.3.3.5 Meßergebnisse 36

2.4 Zusammenfassung und Einordnung des neuen Verfahrens 38

3 Selektion und A/D–Wandlung für Bandpaßsignale **40**

3.1 A/D–Wandlung für Bandpaßsignale 40

3.2 Der konventionelle $\Sigma\Delta$–A/D–Wandler 42

3.2.1 Das Prinzip des Noise Shaping 43

3.2.2 S/N–Gewinn durch Dezimation 44
3.2.3 Noise Shaping für Bandpaßsignale 47
3.2.4 S/N–Gewinn für den Bandpaßwandler 47
3.3 Entwurf des Analogteils 50
3.3.1 DCF77–BP–ΣΔ–A/D–Wandler 57
3.4 Dezimationsfilter für den ΣΔ–Bandpaßwandler 59
3.4.1 Stabilitätstests für Frequenzabtastfilter 60
3.4.2 Stabilität von Elementarfilterstrukturen 62
3.4.3 Bestimmung geeigneter Bit–Breiten in der Kaskade 67
3.4.3.1 Bestimmung der Bit–Breite der einzelnen Register 67
3.4.4 Entwurfsbeispiel 71
3.4.4.1 Realisierung mit parallelen Elementarfiltern 72
3.4.4.2 Realisierung mit Halbbandfilter und Elementarfiltern . . . 73
3.4.4.3 Realisierung mit CBPC–Filtern 74

4 Universelle Demodulatoren 77
4.1 Inkohärente Abtastempfänger für Bandpaßsignale 77
4.1.1 Quadraturüberlagerungsempfänger 79
4.1.2 Die Bandpaßabtastung (Integer–Band–Sampling) 82
4.1.3 Der Quadraturabtastempfänger 83
4.1.4 Der Hilbert–Abtastempfänger 85
4.1.4.1 Hilbert–Abtaster nullter Ordnung 86
4.1.4.2 Hilbert–Abtaster höherer Ordnung 87
4.1.4.3 Hilbert–Abtastempfänger mit Unterabtastung 91
4.1.5 Vergleich der Abtastempfänger 93
4.2 CORDIC–Demodulator 93
4.2.1 Grundlagen des CORDIC–Prinzips 93
4.2.2 Funktionale Beschreibung des Koordinatentransformators 94
4.2.3 Mögliche Hardware–Realisierung des CORDIC–Algorithmus 98
4.2.4 Hilbert–Integer–Band–Aufteilung für BP–ΣΔ–Wandler 102
4.2.5 Vergleich verschiedener Demodulationsprinzipien 103
4.3 Optimalstrategien zur inkohärenten FM–Demodulation 104
4.3.1 Abschätzung zur Bitfehlerwahrscheinlichkeit 106
4.3.2 Strategien zur Optimaldemodulation für FM–LW–Signale 108
4.3.3 Gedächtnisbehaftete Demodulation 110
4.4 Kohärente Empfänger für Bandpaßsignale 113
4.4.1 Linearer Phasenregelkreis 115
4.4.2 Die digitale PLL 117
4.4.3 Costas Regelschleife 119
4.5 Vergleich der realisierten Empfängerschaltungen 123

5 Verschlüsselung und Fehlerkorrektur 131
5.1 Fehlerkorrektur für DCF77 132
5.1.1 Protokollfeldauswertung als Wiederholungscode 133
5.1.2 Erweiterung der Fehlererkennung zum Hamming–Code 133
5.1.3 Fehlerschutz für das gesamte Minutentelegramm 136

5.1.3.1 Codierungsschranken 138
5.1.3.2 Auswahl des Codes für das gesamte Minutentelegramm . . 139
5.1.3.3 Codierung 139
5.1.3.4 Decodierung 140
5.1.4 Vergleich der verschiedenen Verfahren 143
5.2 Personenrufsystem für Langwelle 146
5.2.1 Beschreibung des Datenformats 147
5.2.2 Das Minutenintervall 147
5.2.3 Fehlerkorrekturschaltung 149
5.2.3.1 Der Maximum Likelihood Sequence Decoder — Viterbi–Decoder 150
5.2.3.2 Sequentielle Decoder 156
5.2.3.3 Algebraische Decoder 158
5.2.3.4 Abschließender Vergleich 161
5.2.4 Verschlüsselungsverfahren für FPGAs 162
5.2.4.1 Strom–Chiffren 162
5.2.4.2 Block–Chiffren 164
5.2.4.3 Vergleich der Verfahren bezüglich der kryptographischen Sicherheit 169
5.2.5 Hardware–Struktur des gesamten Decoders 172
5.2.5.1 Blockstruktur des Decoder–FPGAs 173

6 Zusammenfassung **175**

Einleitung

Durch den Einsatz modernster Mikroelektronik ist es möglich geworden, kleine, kompakte, verlustleistungsarme und störsichere Datenempfänger zu realisieren. Als Anwendungsbeispiele solch einer drahtlosen Datenübertragung können hier sowohl unidirektionale Übertragungssysteme wie Funkuhren oder Personenrufsysteme als auch bidirektionale Systeme wie Mobilfunk oder Satellitenkommunikationssysteme genannt werden. Dabei ist nicht nur eine große Produktvielfalt (Diversifikation), sondern auch eine drastische Verkürzung der Entwicklungszeit zu beobachten [Spe94]. Für eine konkurrenzfähige Empfängerentwicklung ist es nötig, die neuartigen Übertragungsverfahren mit komplizierter Signalisierung, Codierung und/oder Verschlüsselung auszurüsten.

In diesem Zusammenhang ist die seit Mitte der 80er Jahre sich stark verbreitende PGA–Technik hilfreich. Diese komplexen programmierbaren Logik–Bausteine (programmierbare Gate Arrays) erlauben die Realisierung recht anspruchsvoller Algorithmen zur Signalverarbeitung und ermöglichen zumeist erst eine „Online" Datenverarbeitung. Ein Mikroprozessor–System ist meist nicht in der Lage, gleich mächtige Algorithmen in der geforderten Zeit in „Software" abzuarbeiten. Die Vorteile des Einsatzes der PGA–Technik in der Empfängerentwicklung sind weitreichend — insbesondere sind aber kurze Entwicklungszeit und Kostenvorteile bei Kleinserien bis 1000 Stück gegenüber einer klassischen ASIC–Lösung bedeutsam. Der Einsatz der PGA–Technik geht dabei weit über das sogenannte „Rapid Prototyping" hinaus, siehe z. B. [Zah90], [Sch91] und [Die91].

Moderne Entwicklungswerkzeuge für PGA–Bausteine bieten teilweise bereits eine automatische Umsetzung der Schaltungsidee in eine PGA–Schaltung für einfache Grundfunktionen an (Schaltungssynthese). Es fehlen aber aussagekräftige (datenbankorientierte) Entscheidungshilfen für den Entwickler welcher Algorithmus vorteilhaft ist. Dies ist z. B. bei der Frage der Fall, welche Schaltung zur Frequenzerzeugung in Bezug auf Platzverbrauch und Leistungsverbrauch günstig in einem PGA–Baustein realisiert werden kann und wie groß der auftretende Frequenz– und Phasenfehler ist.

Weiterhin ist festzustellen, daß die PGA–Gatterkomplexität zwar schon im Bereich von 10.000 Gatteräquivalenten eines NAND–Gatters mit zwei Eingängen liegt, daß aber dennoch bei Unkenntnis eines günstigen Algorithmus eine konventionelle PGA–Implementierung noch unmöglich sein kann. So schlägt z. B. eine konventionelle arctan(Q/I)–Berechnung fehl, während bei Kenntnis des CORDIC–Algorithmus eine PGA–Implementierung der arctan(Q/I)–Funktion durchaus möglich ist.

Hier setzt die vorliegende Arbeit an: Es werden leistungsfähige Algorithmen zur Signalverarbeitung im Hinblick auf eine günstige PGA–Implementierung untersucht.

Da am Institut für Datentechnik bereits seit etwa zwei Jahrzehnten in vier Dissertationen ([Kal80], [Ber85], [Wie88] und [Bel89]) Langwellenempfänger untersucht wurden, beschränkt sich diese Arbeit auf Problemstellungen aus dem Bereich der digitalen

Abtastempfänger für Langwellen–Datensignale, in dem komplexe Algorithmen noch wesentliche Verbesserungen bringen können. Die Erkenntnisse sind nicht nur bei Lanwellenempfängern sondern auch bei Empfängertypen anderer Frequenzbereiche anwendbar.

Die Realisierung von universellen Abtastempfängern umfaßt eine Vielzahl von Aufgabenstellungen. Diese reichen von klassischen, nachrichtentechnischen Problemstellungen, wie Antennenauswahl und Antennencharakteristik, über aufbautechnische Fragestellungen, wie Modularisierung, Schnittstellenfindung und Schnittstellendefinition, bis hin zu informationstechnischen Fragen der Speicherung/Komprimierung der über Langwelle ausgestrahlten (Wetter–)Bilddaten.

Der Schwerpunkt liegt hier jedoch bei den Algorithmen für universelle Abtastempfänger und auf besonders effizienten PGA–Realisierungen. Im einzelnen werden zunächst im ersten Kapitel das grundlegende Signalmodell und die Demodulationsverfahren besprochen. Ebenso werden Beurteilungskriterien für PGAs behandelt. Das zweite Kapitel befaßt sich mit der Fragestellung einer geeigneten Frequenzerzeugung. Hier werden analoge, rein digitale und ein neuartiges hybrides Verfahren untersucht. Kapitel drei behandelt geeignete Selektionsverfahren für Bandpaßsignale und deren Analog/Digital–Wandlung. Insbesondere wird eine neue Klasse von Frequenzabtastfiltern und ein für Langwellenempfänger optimierter Bandpaß–$\Sigma\Delta$–Wandler vorgestellt. Im Kapitel vier werden verschiedene universelle Demodulationsprinzipien auf eine PGA–Implementierung hin untersucht. Ausführlich wird hierbei das Verfahren der Hilbert–Abtastung mit CORDIC–Demodulation beschrieben (das auch schon patentiert werden konnte [MBH92]). Das Kapitel schließt mit einer vergleichenden Untersuchung realisierter Empfängertypen. Kapitel fünf befaßt sich mit zukünftigen Codierschemata für die Langwellendatenübertragung. Zunächst wird erläutert, wie das zur Zeit eingesetzte Funkuhr–Protokoll formatkompatibel durch eine Fehlerkorrektur verbessert werden kann. Anschließend wird ein Personenrufsystem (Pagingsystem) vorgestellt, das konsequent die Vorteile der Langwellentechnik bei der unidirektionalen Datenübertragung benutzt. Die PGA–Implementierung des Paging–Decoders wird beschrieben. Zusammenfassung und Anhang schließen die Arbeit ab.

Kapitel 1

Grundlegende Betrachtungen

Im folgenden Kapitel werden einige grundlegende Betrachtungen zu der Realisierung komplexer Algorithmen für digitale Empfänger in PGA–Technik durchgeführt. Zunächst werden die für LW–Sender üblichen Modulationsarten und typische Sendesignale erläutert, dann folgt in Abschnitt 1.3 eine Beurteilung verschiedener PGA–Architekturen im Hinblick auf ihre Eignung zur Realisierung komplexer Algorithmen.

1.1 Darstellung von Bandpaß–Signalen durch I– und Q–Phase

Bandpaß–Signale lassen sich vorteilhaft durch äquivalente Tiefpaß–Signale beschreiben, die dann im allgemeinen eine komplexe Impulsantwort besitzen. Daher ist es vorteilhaft, die Bandpaß–Zeitsignale unter Bezug auf das Tiefpaß–Signal im Komplexen zu berechnen und anschließend eine Realteilbildung vorzunehmen. So ergibt sich für ein beliebiges Bandpaß–Zeitsignal folgende Darstellung:

$$s(t) \;=\; \mathcal{R}e\left\{s_T(t)e^{j(2\pi f_0 t+\phi_0)}\right\} \tag{1.1.1}$$

Wobei ϕ_0 eine beliebige Anfangsphase beschreibt und $s_T(t)$ im allgemeinen komplex ist. Liegt ein symmetrisches Bandpaß–Signal vor, so vereinfacht sich die Darstellung für reelle $s_T(t)$ zu

$$s(t) \;=\; s_T(t)\cos(2\pi f_0 t + \phi_0)$$

Um mit diesem Signalmodell beliebige AM–, FM– und PM–Signale beschreiben zu können, wird die komplexe Funktion $s_T(t)$ folgendermaßen definiert:

$$s_T(t) \;=\; A(t)e^{j\Delta\phi(t)}$$

Wobei $A(t)$ den Anteil der AM–Modulation und $\Delta\phi(t)$ den Anteil der FM– und PM–Modulation beschreibt.

Das **amplitudenmodulierte Signal** $A(t)$ setzt sich aus einem konstanten Faktor A_0 und einem dem zu übertragenden Sendesignal $s_{\mathrm{AM}}(t)$ proportionalen Anteil zur Amplitude des Trägers zusammen:

$$A(t) = A_0 + s_{\mathrm{AM}}(t) \tag{1.1.2}$$

Charakterisiert wird die AM durch den Modulationsindex m.

$$m = \frac{s_{\mathrm{AM}}(t)|\max}{A_0} \leq 1 \tag{1.1.3}$$

Bei der **Frequenz– bzw. Phasenmodulation** ist das zu übertragende Sendesignal proportional der Frequenz– bzw. Phasenauslenkung des Trägersignals bei konstanter Trägeramplitude A_0. Definiert werden noch die maximale Frequenz– und Phasenabweichung (Phasenhub) bei der Trägerfrequenz f_0 und der Modulationsindex h [AAS86, S. 23-26]:

$$\begin{aligned} \text{Phasenhub:} \quad \Delta\Phi &= \Delta\phi(t)|_{\max} \\ \text{Frequenzhub:} \quad \Delta F &= \frac{1}{2\pi}\frac{d\Delta\phi(t)}{dt}\Big|_{\max} \end{aligned} \tag{1.1.4}$$

$$\text{Modulationsindex:} \quad h = \frac{2 \cdot \Delta F}{f_s} \tag{1.1.5}$$

f_s beschreibt hierbei die maximale Symbolgeschwindigkeit in Baud einer digitalen Übertragung.

Von einem **universellen Empfänger** wird gefordert, daß er mit gleichhoher Qualität das gesendete Signal aus diesen AM–, PM– und FM– modulierten Signalen zurückgewinnen (demodulieren) kann. Daher scheiden viele Empfangsprinzipien aus, die lediglich zum Empfang eines Signaltyps geeignet sind. So der Begrenzer–Diskriminator (FM/PM), Flankendemodulator (FM/PM), Geradeausempfänger (AM) und der Überlagerungsempfänger (AM) ([Lü88, S. 219, S. 237], [ZB87, S. 491-495, S. 526-529], [Wie88, S. 29-31]). Die universellen Empfänger lassen sich in solche mit **kohärenten** und **inkohärenten** Prinzipien aufteilen.

Bei einem *kohärenten* Empfänger besteht die Hauptaufgabe darin, bei bekannter Trägerfrequenz f_0 die unbekannte Trägerphase ϕ_0 zu bestimmen. Hier werden Phasenregelkreise (Phase–Locked Loop PLL, Costas Loop CL, siehe Abschnitt 4.4) eingesetzt, um der sich nur sehr langsam ändernden Ruheträgerphase zu folgen.

Bei einem *inkohärenten* Empfänger ist ebenso die Trägerfrequenz f_0 genau bekannt, hingegen die Phasenlage ϕ_0 nicht. Dies führt, wenn man die Realteilbildung entsprechend Gleichung (1.1.1) ausführt, zu zwei separaten Ausdrücken:

$$\begin{aligned} s(t) &= A(t)\cos\left(2\pi f_0 t + \Delta\phi(t) + \phi_0\right) \\ &= \underbrace{A(t)\cos(\Delta\phi(t) + \phi_0)}_{\text{I–Phase}}\cos(2\pi f_0 t) \; - \underbrace{A(t)\sin(\Delta\phi(t) + \phi_0)}_{\text{Q–Phase}}\sin(2\pi f_0 t) \end{aligned} \tag{1.1.6}$$

Die Anteile mit $\cos(2\pi f_0 t)$ werden **I–Phase** genannt, da sie in Phase zum Träger sind, während die **Q–Phase** in Quadratur [Lü88, S. 107] zum Träger liegt.

Sollen I– und Q–Phase zur Demodulation benutzt werden, so ist es die Aufgabe des Eingangsteils eines Empfängers, eine Aufspaltung des Signals $s(t)$ in I– und Q–Phase

durchzuführen. Dies ist durch Mischen mit Signalen einer Mischfrequenz f_m der Form $2\cos(2\pi f_m t)$ und $-j2\sin(2\pi f_m t)$ (Quadraturüberlagerung) oder durch eine um $T/4$ verschränkte Abtastung (Quadraturabtastung) mit anschließender Filterung zu erreichen. Im folgenden wird gezeigt, wie durch die Hilbert–Transformation direkt eine I– und Q–Phase ohne Mischung und Filterung entsteht. Dies ist Ausgangspunkt für die in Abschnitt 4.1.4 vorgeschlagene Hilbert–Abtastung.

1.1.1 Klassische Erzeugung von I– und Q–Phase durch die Hilbert–Transformation

Nach dem Mischen und anschließender Bandpaß–Filterung ist beim konventionellen Quadraturüberlagerungsempfänger, siehe z. B. [Lü88, S. 112] oder [Wie88, S. 8], eine I– und Q–Phase vorhanden. Bildet man nun aus der I– und Q–Phase durch Addition ein komplexes Signal, so hat das resultierende Spektrum keine Anteile für negative ω. Das komplexe Signal $I + jQ$ ist **analytisch** [Sch84, S. 477], [Lü88, S. 107]. Eine Kombination von $I - jQ$ führt auf ein Signal, dessen Spektrum für positive Frequenzen identisch Null ist. Dieses soll in der Folge als **konjugiert komplexes analytisches** Signal [KK89, S. 131] bezeichnet werden †, siehe Bild 1.1.2. Es wird mit *konjugiert komplex* bezeichnet, da $[I(t) + jQ(t)]^* = I(t) - jQ(t)$ gilt. Hierbei ist zu beachten, daß die Eigenschaft „konjugiert komplex" sich auf die *Zeitfunktion* bezieht, während „analytisch" die Eigenschaft der *Spektralfunktion* beschreibt, daß keine negativen Frequenzanteile auftreten.

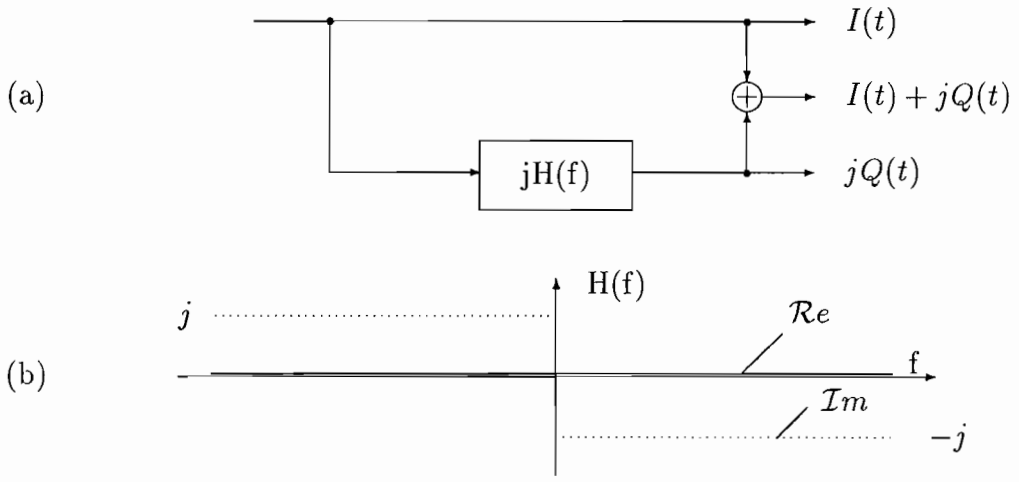


Bild 1.1.1 Der Hilbert–Transformator. (a) Schaltbild. (b) Die spektrale Charakteristik von $H(f)$.

Sucht man auf direktem Weg ein Filter, welches aus einem reellen Signal ein analytisches Signal bildet, so bietet sich hierfür der Hilbert–Transformator an, siehe Bild 1.1.1 und Bild 1.1.2. Die Addition des mit $jH(f)$ multiplizierten Eingangssignals $S(f)$ und des Eingangssignals selbst führt zu dem gesuchten analytischen Signal.

† $I(t) - jQ(t)$ ist auch ein analytisches Signal, da $I(t) - jQ(t)$ auch komplex differenzierbar ist [BS84, S. 518]. Der Begriff analytisches Signal ist aber schon für $I(t) + jQ(t)$ vergeben.

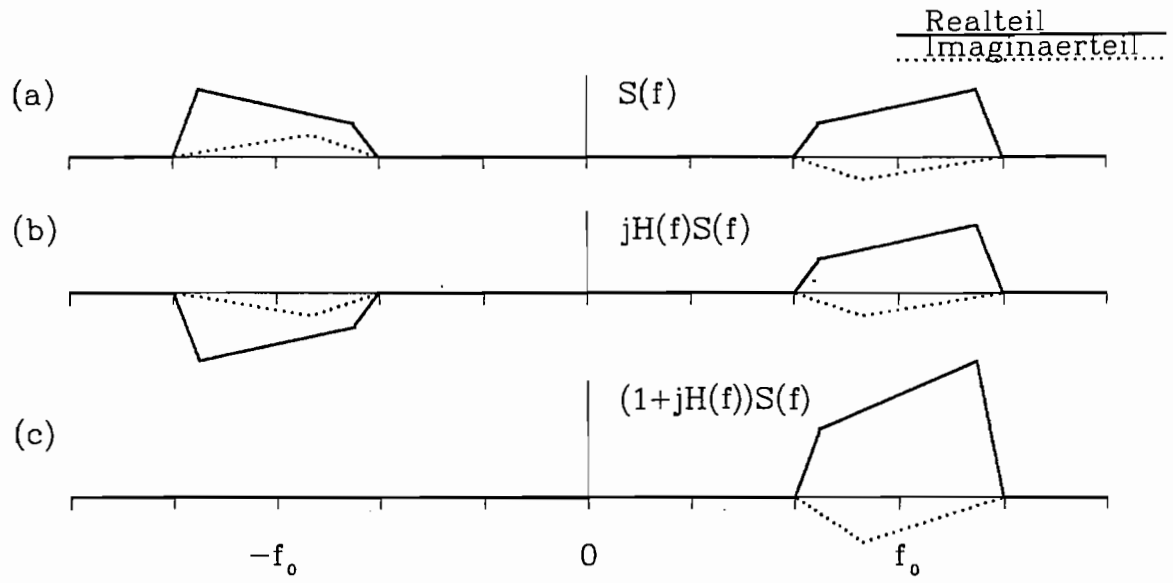


Bild 1.1.2 Bildung eines analytischen Signals mit Hilfe eines Hilbert–Transformators.

Die zugehörige Impulsantwort und Übertragungsfunktion des Hilbert–Transformators ergeben sich durch Berechnung mit Hilfe der Definition der Fourier–Transformation zu [Mar82, S. 125]

$$h(t) = \frac{1}{\pi t} \quad \circ\!\!-\!\!\bullet \quad H(j\omega) = -j\gamma(\omega) = \begin{cases} j & -\infty < \omega < 0 \\ -j & 0 < \omega < \infty \end{cases} \tag{1.1.7}$$

Eine Realisierung ist nur näherungsweise möglich. Klassische Näherungen werden z. B. in [Her69a], [Her69b], [SH90, S. 168-174] und [OS92, S. 681] besprochen.

1.1.2 Kohärente Demodulation

Ist die Trägerphase ϕ_0 genügend genau bestimmt, so kann eine einfache Demodulation durch Multiplikation und TP–Filterung erreicht werden. Bei der AM–Demodulation multipliziert man mit $2\cos(\omega_0 t + \phi_0)$ und bei PM– bzw. FM–Demodulation mit $-2\sin(\omega_0 t + \phi_0)$. Es ergibt sich
AM:

$$\begin{aligned} A(t)\cos(2\pi f_0 t + \phi_0) \cdot 2\cos(2\pi f_0 t + \phi_0) &= \underbrace{A(t)}_{\text{Tiefpaß–Signal}} + A(t)\cos(4\pi f_0 t + 2\phi_0) \\ s_{\mathrm{AM}}(t) &= A(t) - A_0 \end{aligned} \tag{1.1.8}$$

PM:

$$\begin{aligned} -2\sin(2\pi f_0 t + \phi_0) \quad \cdot \quad & \cos(2\pi f_0 t + \phi_0 + \Delta\phi(t)) \\ = \quad & \underbrace{\sin(\Delta\phi(t))}_{\text{Tiefpaß–Signal}} \quad + \cos(4\pi f_0 t + 2\phi_0 + \Delta\phi(t)) \qquad (1.1.9) \\ \sin(\Delta\phi(t)) \quad \approx \quad & \Delta\phi(t) \qquad (1.1.10) \\ s_{\mathrm{PM}}(t) \quad = \quad & \frac{1}{h}\Delta\phi(t) \qquad (1.1.11) \end{aligned}$$

FM:

$$s_{\mathrm{FM}}(t) = \frac{1}{h}\frac{d \quad \Delta\phi(t)}{d\,t} \qquad (1.1.12)$$

1.1.3 Demodulation mit Hilfe von I– und Q–Phase

Die Demodulation von AM–, PM– und FM–Signalen gestaltet sich besonders einfach, wenn I– und Q–Phase vorliegen. Für die verschiedenen Modulationsverfahren ergibt sich im einzelnen:

AM:

$$s_{\mathrm{AM}}(t) = \sqrt{I^2 + Q^2} - A_0 \qquad (1.1.13)$$

PM:

$$s_{\mathrm{PM}}(t) = \frac{1}{h}\left(\arctan\left(\frac{Q}{I}\right) - \phi_0\right) \qquad (1.1.14)$$

FM:

$$s_{\mathrm{FM}}(t) = \frac{1}{h}\frac{d \quad \arctan\left(\frac{Q}{I}\right)}{d\,t} \qquad (1.1.15)$$

oder in Form der „normierten Produktregel“

$$s_{\mathrm{FM}}(t) = \frac{1}{h}\frac{I\dot{Q} - Q\dot{I}}{I^2 + Q^2} \qquad (1.1.16)$$

1.2 Beschreibung der LW–Datensignale

Die auf der Langwelle strahlenden Sender† DCF77, 54, 37 und 49 befinden sich in Mainflingen, 25 km südöstlich von Frankfurt a. M. und besitzen alle eine Sendeleistung von 50 kW, siehe Tabelle 1.2.1. Aus der Frequenzliste für alle LW–Sender (siehe z. B. [Hil93, S. 28]) entnehmen wir, daß die Sendefrequenzen typischer Langwellendatensender im

† DCF kennzeichnet sowohl Ort (D=Deutschland; F Nähe zu Frankfurt a. M.) als auch den Frequenzbereich (C=Langwelle) [Het87].

Bereich 50-150 kHz liegen und einen minimalen Abstand von 5-6 kHz auf dem Frequenzband aufweisen. Somit können mit Hilfe des „Zeitgesetzes der Nachrichtentechnik“ ($B \cdot T = 1$ [Kü74, S. 165]) bereits Rückschlüsse auf die mögliche Schrittgeschwindigkeit bzw. Baud–Rate, die für LW–Sender erreichbar ist, durchgeführt werden. Ein in Darmstadt aufgenommenes typisches Spektrogramm (siehe Bild 1.2.1(a)) zeigt, daß die jeweiligen LW–Datensignale sehr schmalbandig sind und es kaum zu spektralen Überlappungen kommt, wie es bei der breitbandigen FM–Übertragung im UKW–Band vorkommt. Dennoch zeigt Tabelle 1.2.1, daß die Modulationsarten variieren und die realisierten Baud–

Bez.	Daten	P in kW	f_0 in kHz	Modulations–index	Baud–Rate	Bit–Rate
DCF77	Zeit	50	77,5	$m_{\mathrm{AM}} = 0,6$	1	
				$m_{\mathrm{PM}} \rightarrow$16PSK	1	$\sum$ =2
DCF37	Bilder	50	117,4	FM:F1C;h=0,15	ca. 2000	2000
DCF49	Text	50	129,1	FM:F1C;h=1,5	200	200
DCF54	Bilder	50	134,2	FM:F3C;h=0,15	ca. 2000	ca. 8000

Tabelle 1.2.1 Charakteristische Werte der DCF–LW–Sender bei Mainflingen.

bzw. Bit–Raten um einen Faktor 2000 differieren, was hohe Anforderungen an einen universellen Empfänger stellt, der alle Datensignale in gleich hoher Qualität demodulieren soll. Die verwendete FM–Modulation ist schmalbandig, so daß wesentliche Spektralanteile der Signale im Bereich der einseitigen Carson–Bandbreite $\Delta F + f_s$ [Lü88, S. 236] liegen. Für diejenige Bandbreite, in der 95% der Signalleistung enthalten sind, ergab die Simulation eine Bandbreite von $0,151 \cdot f_s$ für DCF37/54 s/w; $0,301 \cdot f_s$ bei DCF37/54 mit 16 Graustufen und $1,001 \cdot f_s$ für DCF49 Sendesignale. Dies zeigt, daß die Carson–Bandbreite für DCF37/54 eine viel zu große Abschätzung liefert. Das Bild 1.2.1(b) zeigt das Ergebnis einer Mittelung über 2000 Spektrogramme mit jeweils 512 zufälligen Bits mit je 32 Abtastwerten im Vergleich zu dem mit Hilfe der Formel in [AAS86, S. 167] berechneten Spektrum für Wetter–FAX s/w bzw. 16 Graustufen und dem 200 Baud Bewag–Signal DCF49.

Der gewählte AM–Modulationsindex für die Funkuhr von 0,6 liegt unterhalb dem maximal möglichen Wert 1,0, da der DCF77–Sender ebenso in seiner ursprünglichen Funktion als Normalfrequenzsender benutzt wird. Spektrale Darstellungen zum FU–Signal sind in [Hil83, S. 84] zu finden.

Die Ausbreitung bzw. Dämpfung der Langwelle über Land und Wasser ist in [Hil93, S. 26-31] ausführlich dargestellt und macht deutlich, daß im Bereich 20-2000 km um den Sender mit einer um 80 dB schwankenden Empfangsamplitude zu rechnen ist [Hil88, S. 257]. Ein universeller Empfänger benötigt deshalb in jedem Fall eine (automatische) Amplitudenregelung (AGC) mit einer relativ hohen Zeitkonstanten, so daß die Signale mit geringer Bit–Rate nicht degradieren. Im folgenden werden die Datensignale im Detail erläutert.

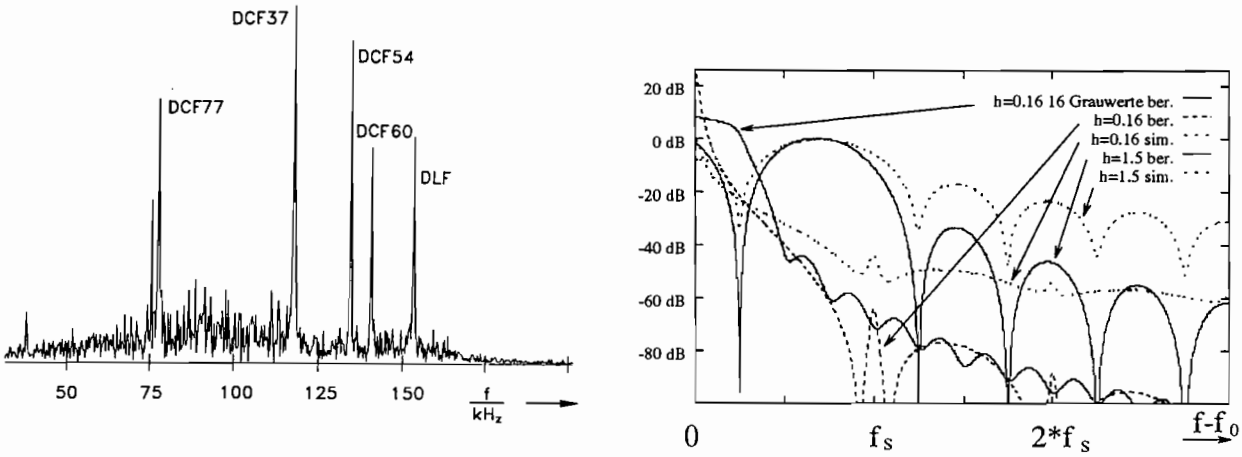


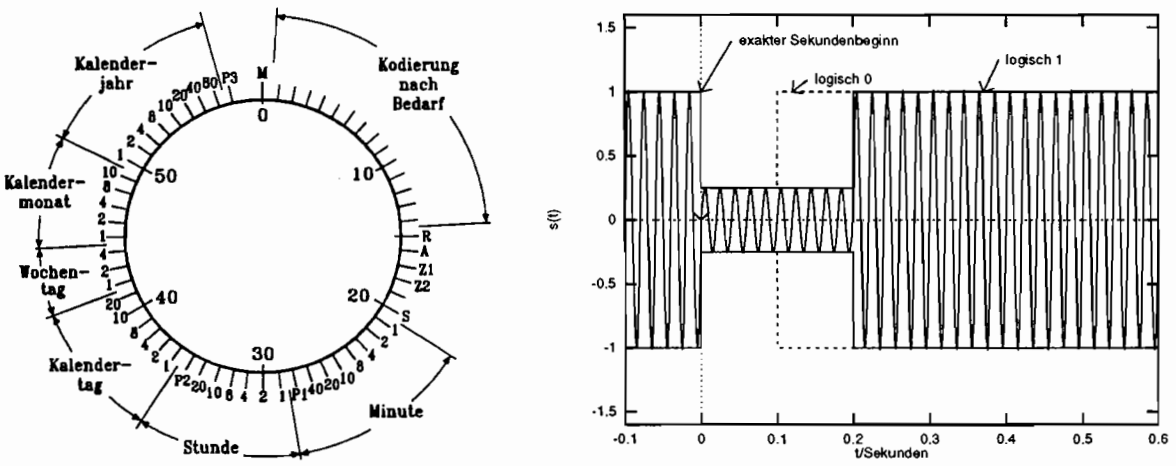
Bild 1.2.1 (a) Gemessenes LW–Spektrum in Darmstadt. (b) Berechnetet bzw. simuliertes Spektrum der Sender DCF37/54 bzw. DCF49.

1.2.1 Der Zeitzeichensender DCF77

Der Zeit– und Normalfrequenzsender DCF77 dient zur Ausstrahlung der gesetzlichen Zeit der Bundesrepublik Deutschland. Seit Juni 1970 wird eine Amplitudenmodulation mit Sekundenmarken durchgeführt. Zu Beginn jeder Sekunde (mit Ausnahme der 59. Sekunde, die der Kennzeichnung des Minutenanfangs dient), wird die Trägeramplitude auf „0“ getastet, so daß in kürzest möglicher Zeit (Rund 1ms [Hil83, S. 259]) die gewünschte 25% Absenkung erreicht wird. Der Start der Absenkung ist der genaue Sekundenbeginn. Die Zeitdaten werden im Bereich 0,1-0,2 der Sekunde gesendet. Ist im Bereich 0,1-0,2 der Sekunde eine Absenkung vorhanden, so entspricht dies einer binären Eins — fehlt die Absenkung, so wurde eine binäre Null übertragen. Faßt man die Absenkung als *einen* Abschnitt auf, so kann statt von einer PAM auch von einer Pulsdauermodulation (PDM) gesprochen werden. Bei PDM besitzt die binäre Eins eine Absenkungsdauer von 0,2 s und die binäre Null eine Dauer von 0,1 s. Das Bild 1.2.2(a) zeigt die Codierung der Zeitdaten innerhalb eines Minutentelegramms. Weitere Informationen zur der Codierung, speziell der Zusatz–Bits 15-20, sind in [Hil93, S. 59-61] zu finden.

Seit 1.1.88 wird neben der PAM zusätzlich eine PSK–Modulation des Trägers mit einer „Pseudo Random Noise“(PN)–Folge der Länge 512 im Bereich 0,2-1,0 der Sekunde durchgeführt, siehe Bild 1.2.3(a). Die PN–Sequenz wird durch ein rückgekoppeltes Schieberegister der Länge 9 erzeugt und besitzt maximale Länge (m–Sequenz). Das Schieberegister führt hierbei eine Polynomdivision mit dem primitiven Polynom $X(z) = 1 + x^5 + x^9$ durch. Da die PN–Folge gleichviel Nullen und Einsen besitzt, bleibt der Mittelwert der Phase unverändert. Es treten $N/2 = 256$ Übergänge (± 1) auf. Die Auslenkung[‡]

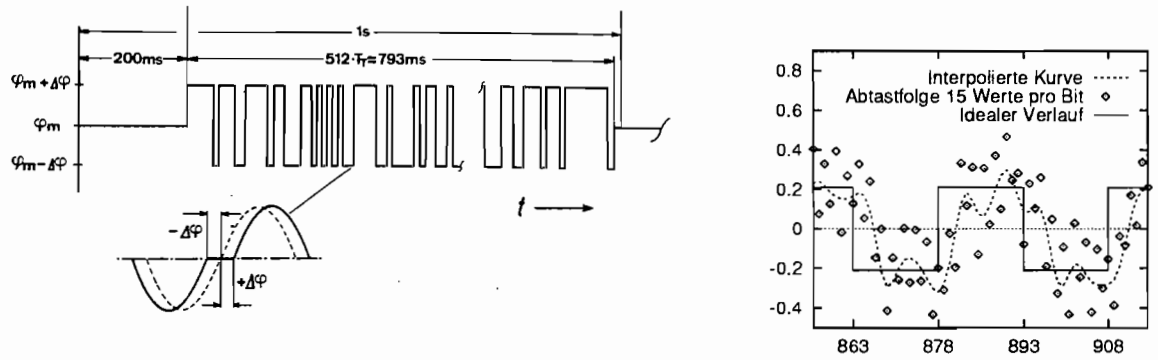
[‡] Beim Vergleich dieser Modulationsart in E_b/N_0 mit anderen Modulationen ist es zutreffender, von einer 360°/24° ≅ 16PSK zu sprechen, als die in der Literatur [Hil93, S. 57] genannte BPSK.



(a) (b)

Bild 1.2.2 (a) Codierschema des Minutentelegramms. (b) Bitinformation der PAM (Trägerfrequenz nicht maßstabsgerecht).

beträgt jeweils $\pm 12°$ und besitzt eine Dauer von 1,55 ms pro PN–Bit. Aufgrund der Bandbreitenbegrenzung durch die LW–Sendeantenne selbst, wird aber, ähnlich wie bei der Amplitudenabsenkung, die Auslenkung erst nach ca. 1/4 der PN–Bit–Dauer erreicht, siehe Bild 1.2.3(b). Dies führt zu einer Verschlechterung der für den genauen Sekundenbeginn zur Verfügung stehenden Bit–Energie der PSK [Wie88, S. 70-78].



(a) (b)

Bild 1.2.3 (a) Idealisierter PN–Verlauf. (b) Gemessener Phasenverlauf [AW91].

1.2.2 Die Wetterbildsender DCF37 und DCF54

Die Langwellensender DCF37 und DCF54 dienen zur Verteilung von Karten des Deutschen Wetter–Dienst (DWD), Offenbach. Neben den direkt vom Meteosat–System§ erstellten Wolkenbildern werden außerdem berechnete Temperaturvorhersagen, Druckverteilung, Windkarten und einige weitere Spezialkarten ausgestrahlt. Die vom Meteosat

§ Die Meteosat–Satelliten sind geostationäre Satelliten in 35800 km Höhe bei 0° bzw. 10° Länge über dem Äquator. Technische Daten: Sendeleistung 200 W, Gewicht 320 kg.

für die verschiedenen Spektralbereiche (6 Bit Auflösung für sichtbaren Bereich; 8 Bit Auflösung für Wasserdampf und Thermal IR) erstellten Bilder werden zunächst mit 166 kBit/s zur Bodenstation (ESOC, Darmstadt) gesendet, danach durch den DWD bearbeitet und über LW wieder ausgestrahlt. Aufgrund der Drallstabilisierung der Satelliten wird eine zeilenweise Abtastung der Erde mit 100 Zeilen pro Minute vorgenommen. Das Teleskop wird nach jeder Umdrehung einen kleinen Winkel gekippt, so daß die Erde nach 25 Minuten durch rund 2500 Zeilen erfaßt ist. Die für die Langwellenausstrahlung aufbereiteten Karten werden mit zwei Zeilen pro Sekunde (FAX-120 Rpm) übertragen. Die theoretisch möglichen 1 Hz und 4 Hz Bilder (60 Rpm bzw. 240 Rpm [fdF90]) werden im Sendeplan des DWD nicht eingesetzt. Die vertikale Auflösung der Bilder ist von der Dauer der Übertragung abhängig. Der größte vorgesehene Zeitraum für ein Bild im Sendeplan beträgt 25 Minuten, was nach Abzug der Dauer für Start- und Stopton eine Auflösung von maximal 3000 Zeilen ergibt [HK92]. Eine genaue Angabe der horizontalen Auflösung ist nicht möglich, da gedruckte Karten über eine optische Zeilenabtastung in einem analogen Verfahren in das Sendesignal umgesetzt werden. Praktische Versuche des DWD bei Übertragung einer Strichmustervorlage ergaben, daß rund 2000 Punkte bei gut 16 Graustufen befriedigend aufgelöst werden können [Nie92].

Jede Bildübertragung wird durch einen drei Sekunden langen Startton der Frequenz 300 Hz eingeleitet. Dann folgt für fünf Sekunden der Synchronsationsbalken. Er besteht aus einem schwarzen oder weißen Balken, der den Anfang jeder Zeile kennzeichnet und auf einer Länge von 1/20 am Anfang der Zeile invertiert übertragen wird. Nach dem Synchronisationsbalken werden keine weiteren horizontalen Synchronisationsinformatio-

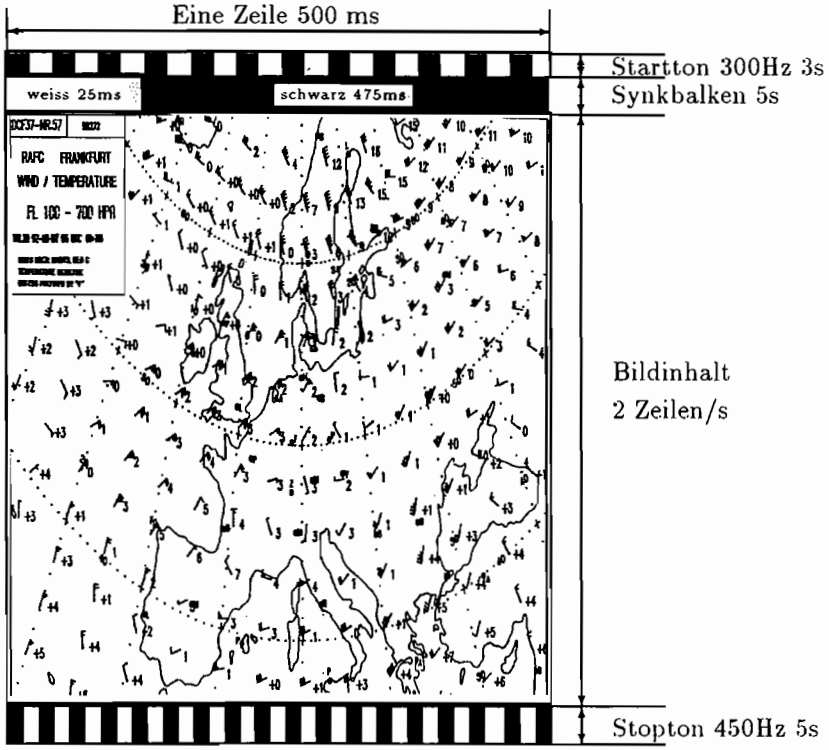


Bild 1.2.4 Das FAX-Format der Wetterbilder.

nen für die restlichen Zeilen übertragen. Am Ende des Bildes wird für fünf Sekunden ein Stopton der Frequenz 450 Hz gesendet. Bild 1.2.4 verdeutlicht den Aufbau des Bildes. Dargestellt ist ein Ausschnitt einer empfangenen Wind/Temperatur–Karte.

1.2.3 Signale des Textsenders DCF54

Von Mitte 92 bis Mitte 93 wurde mit dem DCF54 eine Feldversuch zum Messen der erreichbaren Bitfehlerwahrscheinlichkeit in und um Berlin durchgeführt [ESB93]. Hier wurde bei 200 Baud und einer 8 Bit ASCII–Codierung nach CCITT5 der bekannt Prüftext „THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 1234567890“ gesendet. Dieser Text hat die gewünschte Eigenschaft, daß alle Buchstaben des Alphabets mindestens einmal enthalten sind.

Seit Mitte 93 dient der Sender den EVUs zur Steuerung verschiedenartiger Funktionen, wie z. B. Tarifumschaltung von Stromzählern, Lampensteuerung und lastabhängige Abschalten von Großabnehmern.

1.3 Kriterien zur Beurteilung von programmable Gate Arrays

Die Field Programmable Gate Arrays (FPGA) gehören zur Klasse der programmierbaren Logik ICs (PLIC). Nach der Definition von ASICs (Anwender spezifischer Schaltkreis) sind PLICs wiederum den ASICs unterzuordnen, da mit PLICs Spezialfunktionen und keine Standardlogik realisiert wird. Dennoch soll die allgemein übliche Trennung zwischen PLICs und ASICs vollzogen werden, die besagt, daß für ein ASIC mindestens ein Prozessschritt in der Halbleiterfertigung nötig ist. Gefertigt werden PLICs in nahezu allen verfügbaren Speichertechnologien (SRAM, EPROM, E^2PROM, Antifuse), wobei vorzugsweise FPGAs in SRAM– oder Antifuse–Technik und CPLDs in EPROM– oder E^2PROM–Technik realisiert werden. Alle PLICs realisieren Logikfunktionen. Unter einer Logikfunktion versteht man das Verknüpfen von Schaltvariablen, wobei eine formale mathematische Sprache mit Symbolen und Regeln zur Anwendung kommt [SW90, S. 468-9], die ausschließlich mit digitalen Pegeln („0“=LOW/„1“=HIGH) arbeitet. Die Realisierungen mit PLICs beschränken sich nicht ausschließlich auf das Abbilden von Eingangsvariablen X^n auf Ausgangsvariablen Y^m (Tabelle, ROM), sondern es können sequentielle Automaten realisiert werden, d. h. die Abbildung $X^n \rightarrow Y^m$ ist von den vorherigen Zuständen abhängig $X_t^n \rightarrow Y_t^m$. Dabei wird mit „verdeckter Logik“ (Burried Logic) gearbeitet, d. h. ein Teil der realisierten Funktionen ist an den Bauteilanschlüssen weder sichtbar noch unmittelbar testbar. Dies erfordert zum Testen einer solchen Schaltung in *Software* umfangreiche Entwicklungswerkzeuge (Simulatoren). Beim Test in *Hardware* werden „Boundary Scan–“ bzw. „Readback“–Verfahren eingesetzt, d. h. alle internen Register sind im Readback–Modus zu *einem* Schieberegister† verschaltet und können ausgelesen und analysiert werden. Bei hochkomplexen PLICs wie den FPGAs gehören „Boundary Scan–“ und „Readback“–Implementierungen bereits zum Standard [Tri93].

† Beim „Boundary Scan“ können lediglich die Ein/Ausgangsregister des ICs ausgelesen werden.

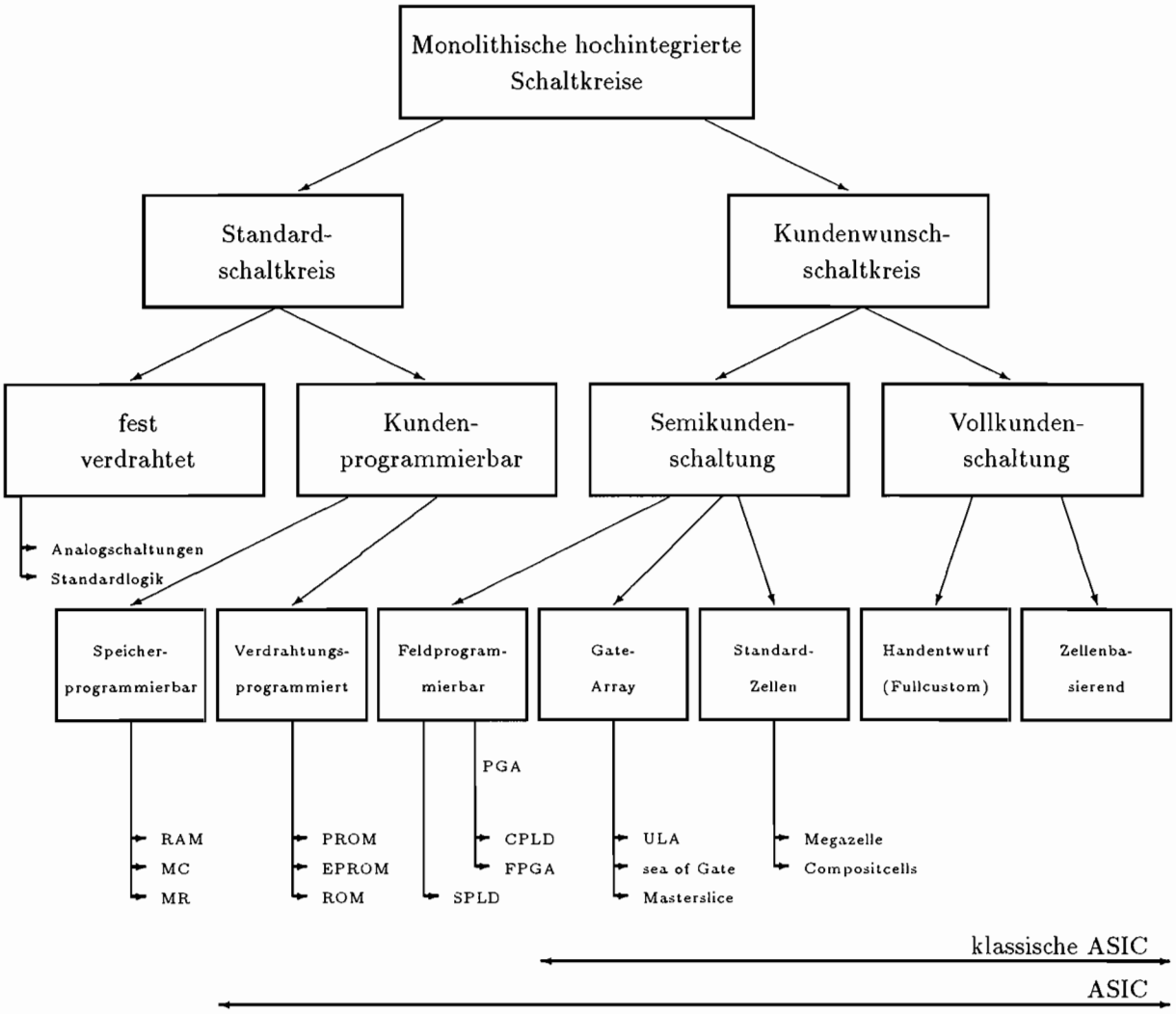


Bild 1.3.1 Einordnung der PLICs.

Zu den PLICs gehören neben FPGAs auch einfache Logikbausteine (SPLD) und die komplexen Logikbausteine (CPLD). FPGAs und CPLDs werden auch unter dem Namen „programmierbare Gate Arrays“ (PGA) zusammengefaßt [DD94]. In Bild 1.3.1 ist eine systematische Einordnung der PLICs dargestellt, die an die Einteilung in [SW90, S. 39] angelehnt ist. CPLDs setzen sich aus einer Mehrfachanordnung von SPLD und einer zusätzlichen, mehrere Bit breite Verbindungsstruktur (Programmable Interconnect Array „PIA“ oder Global Routing Pool „GRP“), die besonders geringe Durchlaufzeiten besitzt, zusammen, siehe Bild 1.3.2(a). FPGAs hingegen besitzen eine feinere Granularität der logischen Grundelemente. Typisch werden 3-5 Eingangsvariablen auf 1-2 Ausgangsvariablen mittels Tabellen (LUT, siehe Bild 1.3.2(b) [Tri93]) oder dedizierten Logikelementen, wie z. B. einer Kombination von Multiplexern [GHB93], abgebildet. CPLD hingegen besitzen die für SPLDs typischen UND/ODER–Strukturen mit 8-10 Eingängen und 3-4 Ausgängen, bei ca. 20 Produkttermen. In der Granularität sehr feine FPGAs werden

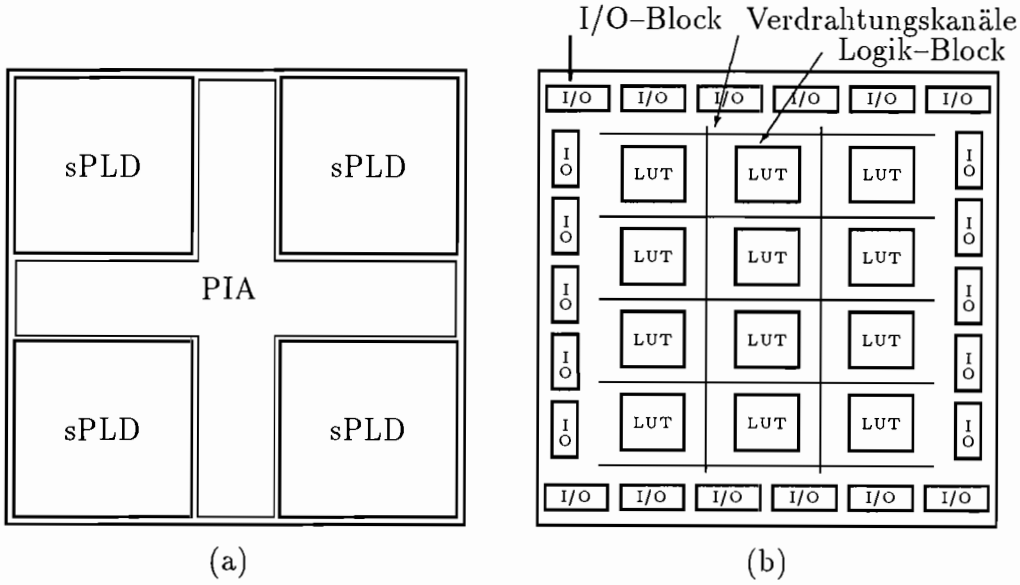


Bild 1.3.2 (a) Typische CPLD–Architektur. (b) Typische FPGA–Architektur.

mit „Sea of Gates"– oder „Pilkington"–Architektur bezeichnet, da sie lediglich ein NAND [Ple90a] oder einzelne „Pass"–Transistoren [CS93] als logische Grundelemente besitzen.

Die Pilkington–Architektur bietet dann Vorteile, wenn Schaltungssynthese auf hohen Abstraktionsebenen mit Hardware–Beschreibungssprachen (HDL, VHDL–1076, REGLAN) durchgeführt werden soll, da so die für Gate Arrays (dies sind ASICs, bei denen Logikelemente in sehr feiner Granularität vorliegt und die Schaltung durch das Aufbringen einer Schicht Metallbahnen realisiert wird) bereits lange verfügbaren Entwurfshilfsmittel eingesetzt werden können. Nachteilig ist bei der sehr feinen Granularität der hohe Verdrahtungsaufwand. Diese Zielrichtung verfolgen klassische ASIC–Hersteller (NEC, Motorola). Es ist unklar, ob die Vorteile dieser feinen Granularität sich in Zukunft durchsetzen werden, da z. B. die NAND–Architektur [DP89] wieder vom Markt genommen wurde. Deshalb sollen diese Bausteine in dieser Arbeit keine weitere Berücksichtigung finden.

Erfolg bzw. Mißerfolg einer Bausteinfamilie ist abgesehen von der Architektur der Bausteine auch ganz wesentlich von der Leistungsfähigkeit der Software–Entwurfshilfen und Synthesemöglichkeiten abhängig. So zeigen Untersuchungen der Logikminimierung mit Hilfe von „Binary Decision Diagramms" (BDD) [Ake78], [Bry86], anhand des in Kapitel 5 beschriebenen Meggit–Decoders, daß statt der von Xilinx–Software benötigten 132 CLBs bei der BDD–Minimisierung lediglich 50 CLBs, d. h. eine Aufwandsreduktion von 62% erreicht werden kann [dR93], [SVGR93].

Nun bleibt zu fragen, warum PLICs seit Anfang der achtziger Jahre und nach den Prognosen [Nag90], [Tec92] bis weit über das Jahr 2000 Wachstumsraten deutlich über 20% besitzen und damit die klassischen ASICs um rund 10% Punkte übertreffen. Die PLICs besitzen nahezu alle Vorteile der ASICs gegenüber einem Schaltungsentwurf mit Standardbausteinen wie

- Verringerung von Platz, Gewicht und Verlustleistung

- Erhöhte Betriebsfrequenz
- Sicherheit gegen Nachbau
- kostengünstige Lösung
- geringe Testkosten

aber nicht die Nachteile der ASICs, sondern

- eine erheblich kürzere Entwicklungszeit (Rapid Prototyping) Zeitfaktor: 3-4
- „In Circuit Reprogrammiebar“[‡], leichte Veränderungsmöglichkeiten
- Wesentliche geringere Einmalkosten (NRE–Kosten), wesentliche Kostenvorteile bei Stückzahlen kleiner 1000

Dennoch bleibt festzuhalten, daß bei stetig steigenden Gatteräquivalenten von PLICs und ASICs, die AISCs ca. um einen Faktor 10 höhere Gatteräquivalente als PLICs besitzen und wesentliche Kostenvorteile bestehen, wenn Stückzahlen größer 1000 gefordert sind.

Für einen Vergleich von CPLDs und FPGAs für Anwendungen von komplexen Algorithmen in der Signalverarbeitung, wie die vorliegende Arbeit untersucht, sind die seit 3/93 verfügbaren Benchmarks der „Programmable Electronics Performance Corporation“ (PREP) [Bur93] hilfreich. Diese Benchmark lagen zu Beginn der vorliegenden Untersuchung (1990) nicht vor, zeigen aber, daß die eigenen vergleichenden Untersuchungen und die Wahl einer mittleren Granularität der PLICs für die zu realisierenden Algorithmen am besten geeignet sind.

PREP ist (Mitte 94) eine Vereinigung von 13 führenden PLIC–Hersteller (Gründer waren Xilinx, Altera und Actel) und Softwarehäusern, die insgesamt 9 verschiedene, PLIC typische Schaltungen als Vergleichstest entworfen haben. Jeder Hersteller führt die 9 Tests selbst durch, indem die Grundstruktur möglichst zahlreich (Repetition Rate „REP“) bei maximaler Taktgeschwindigkeit im jeweils größten Baustein der Familie realisiert wird. Für die Algorithmen zur digitalen Signalverarbeitung erscheinen besonders die Tests „5“ (4 Bit Multiplizierer gefolgt von einem 8 Bit Akkumulator) und „6“ (16 Bit Akkumulator) hilfreich. Die anderen Tests beschreiben spezielle Decoderschaltungen („Glue Logic“). Anhand dieser 2 Tests zeigt das Diagramm 1.3.3(a) die gemittelten Meßergebnisse für die 1990 verfügbaren Bausteinarchitekturen (Actel A_2, Xilinx $\times_1$ und Altera o_1)[§]. Deutlich wird, daß die FPGA–Struktur zu bevorzugen ist, wenn hohe Komplexität gefordert ist. Da Actel nur „Antifuse“ basierte FPGAs anbietet, d. h. Bausteine, die nur einmal programmiert werden können, wurden die in den Kapiteln 2-5 durchgeführten Untersuchungen mit Xilinx–FPGAs durchgeführt. Neuere FPGA–Familien, wie Altera 81000 seit

[‡] Unter „In Circuit Reprogrammiebar“ versteht man die Möglichkeit, in einer vollständig aufgebauten Schaltung durch Programmierung des PLICs wesentliche Veränderungen am Logikverhalten der Schaltung vornehmen zu können, ohne daß Veränderungen an dem Schaltungsaufbau selbst vorgenommen werden.

[§] Zum damaligen Zeitpunkt waren vergleichbare Vorgängertypen: Actel 1020, MAX5128 und Xilinx 3090-100 verfügbar.

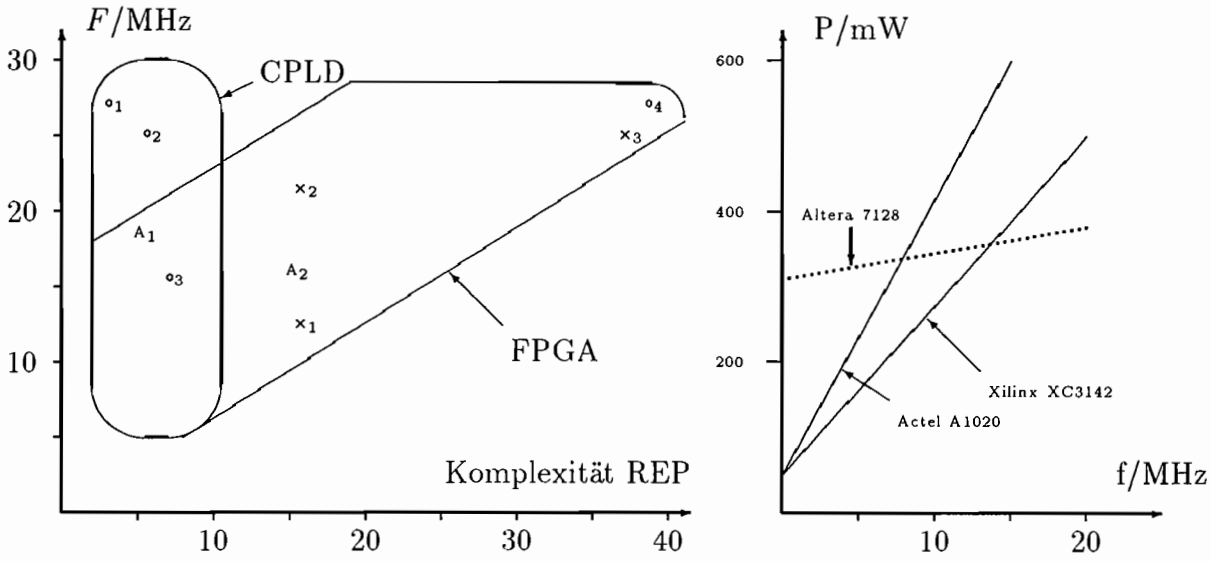


Bild 1.3.3 (a) PREP–Benchmark Mittelung von Test 5+6. A=Actel, o =Altera, x=Xilinx: o_1 =7128; o_2 =7192; o_3 =7256; o_4 =8188-3;A_1 =A1225;A_1 =A1240; x_1 =3090-125; x_2 =3190-3; x_3 =4010-5; (b) Vergleich des Leistungsverbrauchs von CPLDs (Altera) und FPGAs (Actel, Xilinx) bei ca. 85% Nutzung der Logikmodule.

1993 und Xilinx mit der Familie XC4000 bieten noch erhebliche Verbesserungen in der Komplexität und Geschwindigkeit, siehe Bild 1.3.3(a) o_4 und x_3.

Die PREP–Benchmarks sind zwar recht hilfreich bei der Beurteilung der zu realisierenden Arithmetik und Verdrahtungskapazität innerhalb der ICs, aber eine Gesamtbeurteilung der PLICs ist noch von weiteren Faktoren abhängig wie

- Leistungsverbrauch
- OnChip–RAM
- Pin zu Pin Laufzeit
- Interner „Tri State"–Busse
- „Readback–" oder „Boundary Scan–"Decoder
- Programmierbarkeit der Anstiegszeit der I/O–Treiber

PREP beabsichtigt weitere System–Benchmarks zu definieren. Ein sehr wesentliches Kriterium bei der Realisierung portabler Systeme/Empfänger ist der Stromverbrauch. So zeigen die Untersuchung in [Hil92b] und [Rie93] die Vorteile der FPGAs bei Taktfrequenzen kleiner 5 MHz, siehe Bild 1.3.3(b).

Abschließend zur Beurteilung von FPGAs und CPLDs bleibt festzuhalten, daß CPLDs bei schneller „Decoder Logik" zu bevorzugen sind, während FPGAs besser zur Realisierung komplexer Algorithmen geeignet sind.

Kapitel 2

Methoden der Frequenzsynthese

Beim Entwurf digitaler Empfänger besteht eine übergeordnete Aufgabe darin, für die verschiedenen Schaltungsteile die zugehörige Taktfrequenz zur Verfügung zu stellen. Insbesondere sollten Abtast–und Mischfrequenz bei einem universellen Empfänger veränderlich sein und eine hohe Auflösung bei einem Minimum an Frequenz– bzw. Phasenrauschen besitzen.

Im folgenden Kapitel wird sich zeigen, daß klassische analoge Verfahren eine zu große Frequenzungenauigkeit und einen zu starken Temperaturdrift besitzen [TS86], [ZB87]. Die direkten digitalen Synthesizer (DDS) wie Frequenzteiler, multiplikative Frequenzerzeugung und der klassische Akkumulator–Synthesizer bieten ein gutes Temperaturverhalten und hohe Frequenzgenauigkeit, aber zumeist einen nicht akzeptablen Frequenz– bzw. Phasenfehler [She90] und zudem einen unakzeptablen Leistungsverbrauch (ca. 1 W).

Der Einsatz eines hybriden Verfahrens bedingt, daß neben dem FPGA weitere, möglichst wenige, analoge Bausteine eingesetzt werden müssen. Es zeigt sich, daß nur mit hybriden Verfahren das geforderte Frequenz– bzw. Phasenrauschen und der Leistungsverbrauch akzeptabel sind. Es werden klassische hybride Verfahren, die sich der PLL–Technik bedienen, untersucht. Dann wird ein neues, Akkumulator basiertes Verfahren vorgestellt, das sich durch geringen Aufwand, kleinen Phasen– und Frequenzfehler und exzellent kurze Einschwingzeit auszeichnet. Meßtechnische Verifikation der Simulationen und ein Vergleich der Verfahren schließen das Kapitel ab.

2.1 Klassische analoge Verfahren zur Frequenzsynthese

Bei den analogen Verfahren sind in erster Linie RLC–Schwingkreise zu nennen, bei denen ein Schwingkreis über ein Verstärkerelement in Mitkopplung zur Dauerschwingung angeregt wird, siehe Bild 2.1.1.

Die Durchstimmbarkeit des Schwingkreises wird meist durch Veränderung der Kapazität erreicht. Dies kann über einen Drehkondensator (mit entsprechendem Antrieb) erfolgen. Weit verbreitet ist die elektrische Steuerung des Kreises über eine Kapazitätsdiode (Varicap), bei der die Sperrschichtkapazität und damit die Resonanzfrequenz durch eine elektrische Vorspannung (Gleichspannung) bestimmt wird, siehe Bild 2.1.1(b).

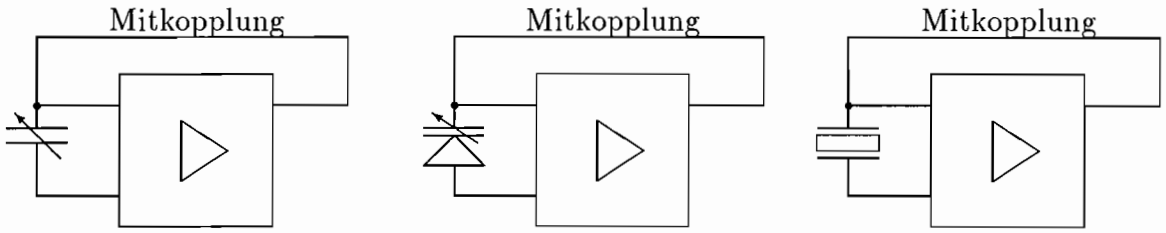


Bild 2.1.1 Analoge steuerbare Frequenzsynthesizer. (a) RLC–Kreis mir Drehkondensator. (b) RLC–Kreis mit Varicap. (c) Quarzoszillator.

Die RLC–Oszillatoren sind folgendermaßen zu bewerten:

\+ Geringer Schaltungsaufwand

\++ Geringe Verlustleistung

\– – Geringe Frequenzgenauigkeit (1% Bauteiltoleranzen)

\– Frequenzinstabilität durch Temperaturdrift und Alterung

Zu erwähnen sind noch die Quarzoszillatoren, die bezüglich Stabilität und Genauigkeit der Frequenz den RLC–Oszillatoren weit überlegen sind, da mechanische Resonanzerscheinungen des Quarzes ausgenutzt werden, die eine sehr große Güte (Bandbreite/Mittenfrequenz) besitzen. Die Quarzoszillatoren können auf der Grundwelle oder durch entsprechende Filterung und Mitkopplung auf Oberwellen schwingen. Im übrigen sind aber die Möglichkeiten einer Durchstimmbarkeit über Parallelkapazitäten über weite Frequenzbereiche mit Quarzoszillatoren *nicht* gegeben, so daß die Quarzoszillatoren nicht weiter betrachtet und beurteilt werden sollen.

2.2 Direkte Digitale Synthesizer

Im folgenden werden drei bekannte DDS–Verfahren auf eine mögliche Realisierung in PGA–Technik hin untersucht. Die verwendeten Definitionen von Phasen– und Frequenzfehler werden am Beispiel des Akkumulatorsynthesizers erläutert.

2.2.1 Direkte Digitale Synthesizer mit Akkumulator

Beim bekannten Akkumulator–Synthesizer wird der Überlauf eines N–Bit–Registers nach vielfacher Addition eines konstanten Summanden M detektiert [Tel90], [Har90], [Qua91]. Für eine symmetrische Takterzeugung kann dieser Überlauf ein Toggle–Flip–Flop (TFF) umschalten, während bei einer unsymmetrischen Takterzeugung, wie sie im folgenden betrachtet werden soll, ein Monoflop, Abtaster etc. getriggert wird, so wie es in Bild 2.2.1(a) dargestellt ist. Einige Anstrengungen wurden unternommen, um die spektrale Reinheit bei sin/cos–Erzeugung zu verbessern [NS91], [Ple89]. Diese Anstrengungen beschränken

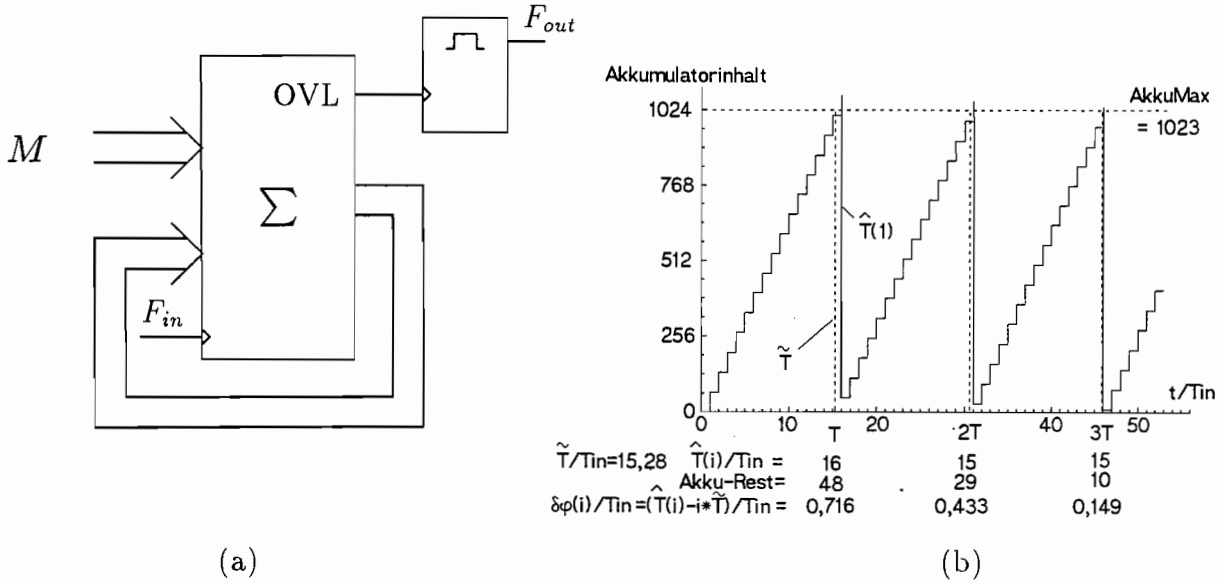


Bild 2.2.1 (a) Unsymmetrische Takterzeugung mit einem Akkumulator. (b) Überlaufverhalten des Akkumulator–Synthesizer.

sich allerdings auf möglichst geringe Nebenzipfel, während die Spektral–Breite des Hauptzipfels, die über den Frequenzfehler Aufschluß gibt, unberücksichtigt bleibt. Dies wird in der Folge analytisch untersucht.

Bei einer Mittelung der Periodendauer über viele[‡] Perioden ergibt sich die Sollfrequenz $\widetilde{F}$ zu

$$\widetilde{F} = F_{\text{out}} = \frac{M}{2^N} F_{\text{in}} \tag{2.2.1}$$

wobei ein N–Bit breiter Akkumulator benutzt wird. Damit ergibt sich eine **theoretische Frequenzauflösung** von

$$\mathrm{D}F = \left(\frac{M+1}{2^N} - \frac{M}{2^N}\right) F_{\text{in}} = \frac{1}{2^N} F_{\text{in}} \tag{2.2.2}$$

Als Beispiel sei das Verhalten des Akkumulators bei $F_{\text{in}} = 1,18$ MHz, $F_{\text{out}} = 77,5$ kHz, $N = 10$, $M = 67$ für die ersten 3 Zyklen in Bild 2.2.1(b) betrachtet.

Deutlich ist zu erkennen, daß die Istfrequenz von der Sollfrequenz abweicht. Es treten nicht die mittlere Periodendauer von $2^N/M = 15,28$, sondern die benachbarten Ganzzahlen [§] $\lfloor 15,28 \rfloor = 15$ und $\lceil 15,28 \rceil = 16$ auf.

Phasen– und Frequenzabweichungen sind unterschiedlich zu bewerten. So ist bei PM–modulierten Signalen auf möglichst geringen Phasenfehler von Mischer, Abtaster etc. zu achten, während bei einer AM– oder FM–Demodulation ausschließlich der Frequenzfehler für die Qualität ausschlaggebend ist.

[‡] Die Anzahl der Perioden, über die gemittelt werden muß, wird in Abschnitt 2.3.3.1 exakt berechnet.

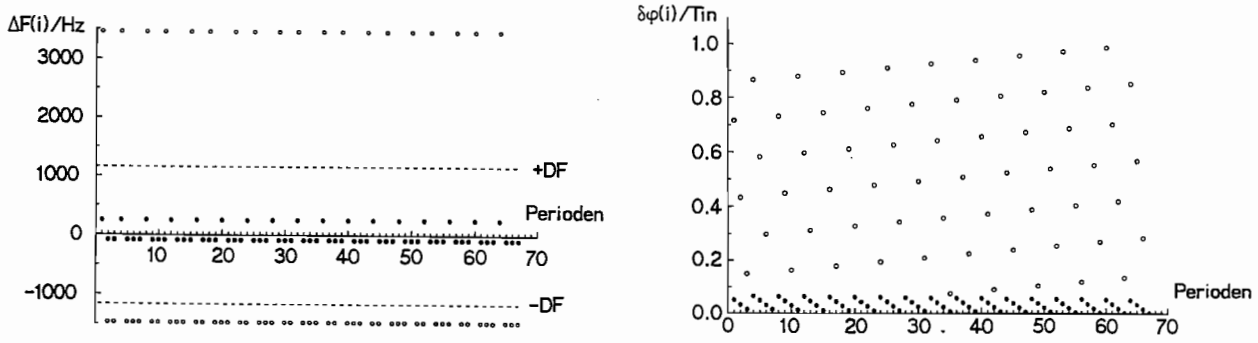
[§] $\lfloor x \rfloor$ steht für die nächst kleinere Ganzzahl, während $\lceil x \rceil$ für die nächst größere Ganzzahl steht.

Die **Frequenzabweichung** $\Delta F(i)$ ergibt sich aus der Differenz zwischen *Momentanfrequenz* ($\widehat{F}(i) = (\widehat{T}(i) - \widehat{T}(i-1))^{-1}$) und *Sollfrequenz* ($\widetilde{F} = 2^N/M \cdot F_{\text{in}}$), wobei jeweils eine Periode der Ausgangsfrequenz betrachtet wird.

$$\Delta F(i) = \frac{1}{\widehat{T}(i) - \widehat{T}(i-1)} - \frac{1}{\widetilde{T}} \tag{2.2.3}$$

Als *mittlerer Frequenzfehler* ΔF wird das Mittel über einen vollständigen Zyklus L bestimmt.

$$\Delta F = \frac{1}{L} \sum_{i=1}^{L} \left| \widehat{F}(i) - \widetilde{F} \right| \tag{2.2.4}$$



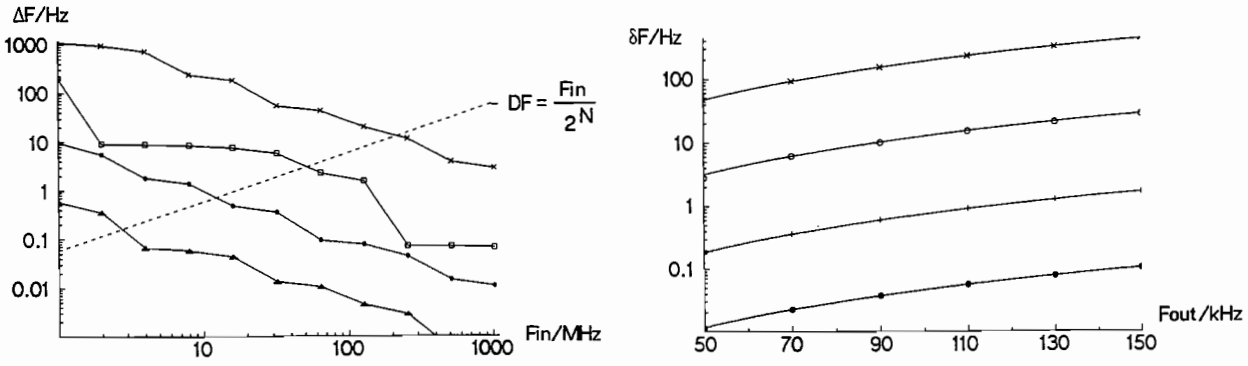
(a) (b)

Bild 2.2.2 (a) Frequenzfehlerverhalten für einen vollständigen Zyklus. (b) Phasenfehlerverhalten für einen vollständigen Zyklus. Ohne o und mit • 4 Bit D/A–Kompensation (Kompensation siehe Abschnitt 2.3.3).

Für unser Beispiel von oben folgt entsprechend Bild 2.2.2(a, unkompensiert) $\Delta F = 2033,7$ Hz. Die gestrichelte Linie gibt die *Frequenzauflösung* von $DF = F_{\text{in}}/2^N = 1156,72$ Hz an. Die Tabelle 2.2.1 verdeutlicht nochmals, warum diese Auflösung als *theoretische Frequenzauflösung*† bezeichnet werden soll, denn der mittlere Frequenzfehler bleibt trotz wachsendem N gleich. Die Anzahl der Perioden mit der Länge T ist in der Tabelle mit $\subset T \supset$ bezeichnet.

Die **Phasenabweichung** ergibt sich aus der Differenz zwischen dem tatsächlichen Umschaltzeitpunkt ($\widehat{\phi}(i) = \widehat{T}(i)$) und dem korrekten Umschaltzeitpunkt ($\widetilde{\phi}(i) = i \cdot \widetilde{T}$), wobei jeweils eine Periode der Ausgangsfrequenz betrachtet wird.

† So wird von Harris in [Har90] die Auflösung ihres 32 Bit breiten Akkumulator–Synthesizer mit 0,01 Hz bei $F_{\text{in}} = 30$ MHz angeben, obwohl der mittlere absolute Frequenzfehler 34,9 Hz für z. B. $F_{\text{out}} = 77,5$ kHz beträgt und somit um mehr als einen Faktor 1000 größer ist!



(a) (b)

Bild 2.2.3 (a) Frequenzfehler für verschiedene M bei festem F_{out}: ($\times$) unkompensiert. ($\square$) D/A–Wandler 4 Bit. ($\bullet$) D/A–Wandler 8 Bit. ($\triangle$) D/A Wandler 12 Bit.($N = 24$; $M = 1270 \cdot 2^i \quad i = 0 \cdots 10$; $F_{\text{in}} = 2^i$ MHz; $F_{\text{out}} = 77{,}5$ kHz) (b) Phasenfehler für verschiedene M bei festem F_{in}: ($\times$) unkompensiert. (o)D/A–Wandler 4 Bit. (+)D/A–Wandler 8 Bit. ($\bullet$)D/A–Wandler 12 Bit. ($N = 17$; $F_{\text{out}} = 50 \cdots 150$ kHz; $M = 500 \cdots 1500$; DF=100 Hz; F_{in}=26,2144 MHz);(Kompensation siehe Abschnitt 2.3.3)

$$\delta\phi(i) \;=\; \widehat{T}(i) - i * \widetilde{T} \tag{2.2.5}$$

Da die so definierte Phasenabweichung keinen Bezug zur Ausgangsfrequenz besitzt, soll die Phasenabweichung normiert werden. Es ergibt sich als *äquivalente Differenzfrequenz* $\delta F(i)$:

$$\delta F(i) \;=\; \frac{\widehat{T}(i) - i \cdot \widetilde{T}}{\widetilde{T}} \cdot F_{\text{out}} \tag{2.2.6}$$

Für unser Beispiel von oben ergibt sich entsprechend Bild 2.2.2(b, unkompensiert) als mittlerer Phasenfehler die *mittlere äquivalente Differenzfrequenz*:

$$\begin{aligned} \delta F \;&=\; \frac{1}{L} \sum_{i}^{L} \frac{\widehat{T}(i) - i \cdot \widetilde{T}}{\widetilde{T}} \cdot F_{\text{out}} \\ &=\; 2497{,}56 \text{ Hz} \end{aligned}$$

Aus Bild 2.2.2(b) wird deutlich, daß der Phasenfehler im gesamten Zyklus kleiner $\widetilde{T}$ bleibt.

Abschließend zur Fehlerbetrachtung sei der (unkompensierte) Phasen– und Frequenzfehler der Synthesizer in den Bildern 2.2.3(a) und (b) betrachtet. Aus Bild 2.2.3(b) wird

Bit–Breite	M	F_{out}	$\subset 15 \supset$	$\subset 16 \supset$	Zyklus	FehlerΔF
N=10	67	77500 Hz	48	19	67	2033,75 Hz
	68	78656,72 Hz	16	1	17	562,48 Hz
Frequenzauflösung DF=		1156,72 Hz				
N=16	4288	77500 Hz	48	19	67	2033,75 Hz
	4289	77518,07 Hz	3088	1201	4289	2018,65 Hz
Frequenzauflösung DF=		18,074 Hz				
N=24	1097728	77500 Hz	48	19	67	2033,75 Hz
	1097729	77500,07 Hz	786448	311281	1097729	2033,69 Hz
Frequenzauflösung DF=		0,071 Hz				
N=32	281018368	77500 Hz	48	19	67	2033,75 Hz
	281018369	77500 Hz	201326608	79691761	281018369	2033,75 Hz
Frequenzauflösung DF=		0,00028 Hz				

Tabelle 2.2.1 Frequenzauflösung für verschiedene Akkumulatorbreiten N bei $F_{in} = 1,18$ MHz und benachbarten Werten M.

deutlich, daß bei linear ansteigender Ausgangsfrequenz F_{out} die oberen Schranken für Phasen– und Frequenzfehler dem Potenzgesetz gehorchen:

$$\Delta F = \delta F \quad = \quad k \cdot \sqrt{F_{out}} \tag{2.2.7}$$

Die Konstante k hängt hierbei von dem Verhältnis F_{in}/F_{out} ab und ergibt sich für dieses Beispiel bei F_{in}=26,2144 MHz zu $k = 7,23$. Weiterhin sei angemerkt, daß der Phasenfehler für alle Frequenzwerte M im allgemeinen sehr nahe an der oberen Schranke liegt, während beim Frequenzfehler auch sehr viel geringere Werte auftreten können.

Aus Bild 2.2.3(a) wird besonders deutlich, daß der Phasen– und Frequenzfehler *nicht* von der Akkumulatorbreite, sondern von dem Verhältnis von Eingangs– zu Ausgangsfrequenz (F_{in}/F_{out}) abhängt. So ist zu erkennen, daß der Phasen– und Frequenzfehler erst ab F_{in} >300 MHz kleiner als die theoretische Frequenzauflösung DF wird. Eine Reduktion des Phasen– und Frequenzfehlers ist somit durch Erhöhung der Taktfrequenz des Akkumulators einfach zu erreichen. Die Werte des Phasenfehlers liegen hierbei geringfügig oberhalb denen des Frequenzfehlers.

Zusammenfassend läßt sich der klassische Akkumulator–Synthesizer folgendermaßen beurteilen: Positiv ist der lineare Zusammenhang zwischen Frequenzwert M und der Ausgangsfrequenz F_{out} sowie der große Frequenzvariationsbereich. Nachteilig bleibt die für geringen Phasenfehler nötige hohe Taktfrequenz und damit verbunden ein hoher Leistungsverbrauch bzw. bei niedriger Taktfrequenz ein erheblicher Frequenz– bzw. Phasenfehler.

2.2.2 Der Frequenzteiler

Die quarzgenaue Frequenz F_{in} wird einem variablen Teiler M zugeführt, an dessen Ausgang die zu erzeugende Frequenz F_{out} entsteht. Die Ausgangsfrequenz ist hierbei auch quarzgenau, d. h. das auftretende Frequenz– bzw. Phasenrauschen ist annähernd Null.

$$F_{\text{out}} = \frac{F_{\text{in}}}{M} \tag{2.2.8}$$

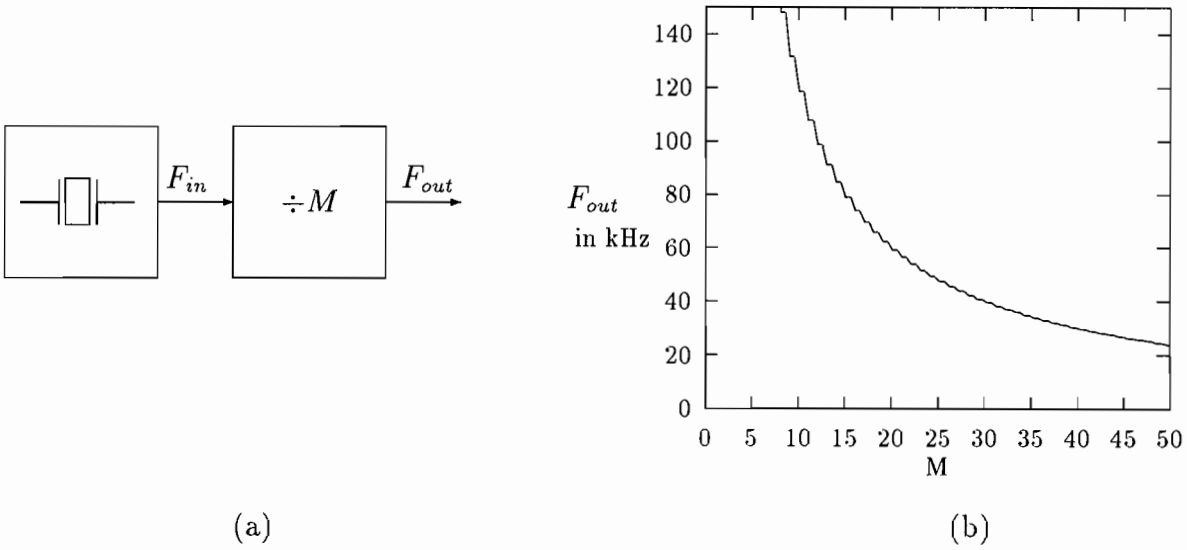


Bild 2.2.4 Frequenzerzeugung durch Frequenzteilung. (a) Schaltung. (b) Nichtlinearer Zusammenhang zwischen M und F_{out}.

Aus Gleichung (2.2.8) wird deutlich, daß der Zusammenhang zwischen Ausgangsfrequenz und Frequenzparameter M *nichtlinear* ist, so daß bei einem linearen Frequenzwechsel jeweils die Division entsprechend Gleichung (2.2.8) ausgeführt werden muß, siehe Bild 2.2.4(b).

Ein größerer Nachteil ist aber: Für eine hohe Auflösung der Frequenz muß die Eingangsfrequenz bei der Frequenzsynthese mit dem Frequenzteiler sehr viel größer als die zu synthetisierende Frequenz sein.

Dies sei an folgendem Beispiel verdeutlicht:

Beispiel 2.1: Mischfrequenzerzeugung

Ziel sei alle Frequenzen $F_{\text{out}} = 50$ kHz $\cdots$ 150 kHz im Frequenzabstand $DF = 100$ Hz zu erzeugen, wobei die Frequenzabweichung kleiner als 10 Hz sein soll — was in etwa dem Frequenzhub für eine Graustufe im WeFAX–Format (siehe Abschnitt 1.2) entspricht — so folgt

$$10 \text{ Hz} = F_1 - F_2 = \frac{F_{\text{in}}}{M} - \frac{F_{\text{in}}}{M+1} \tag{2.2.9}$$

$$\rightarrow F_{\text{in}} = \frac{1}{\frac{1}{F_2} - \frac{1}{F_1}} \tag{2.2.10}$$

$$= \frac{1}{\frac{1}{149{,}99\ \text{kHz}} - \frac{1}{150\ \text{kHz}}} = 2250\ \text{MHz} \tag{2.2.11}$$

Die erforderliche Bit–Breite des Teilers ergibt sich zu $\lceil F_{\text{in}}/F_{\text{out}} \rceil$=16 Bit.

Die Beurteilung des Frequenzteilers läßt sich folgendermaßen zusammenfassen:

++ hohe Stabilität, geringes Phasenrauschen

\+ geringer Schaltungsaufwand

\− nichtlinearer Zusammenhang zwischen F_{out} und M

−− hohe Taktfrequenz, Leistungsverbrauch

\− geringe Auflösung

−− kein konstantes Frequenzraster möglich

2.2.3 Multiplikative Frequenzerzeugung

Bei der multiplikativen Frequenzerzeugung wird die Eingangsfrequenz F_{in} in jeder Stufe mit $n/10$ mit $n \in [0..9]$ multipliziert, indem jeweils n Takte aus 10 Takten „ausdecodiert“ werden. Dieses Prinzip ist in dem TTL–Baustein 74LS97 bzw. 74LS167 realisiert und ermöglicht eine lineare binäre bzw. BCD Einstellung der Ausgangsfrequenz. Die Bausteine sind als Module in den Software–Entwurfshilfen der FPGA–Hersteller meist enthalten [Nie93].

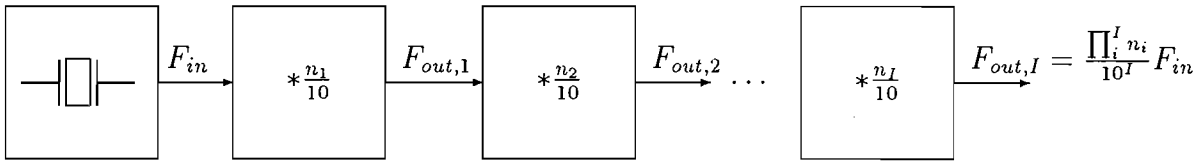


Bild 2.2.5 Frequenzerzeugung durch I fache Multiplikation.

Es tritt bei dieser Art der DDS in jeder Stufe ein Phasen– bzw. Frequenzfehler auf, der bis $\frac{T_{\text{in},i}}{2}$ groß sein kann, was, vor allem in der letzten Stufe I, zu einem erheblichen Fehler führt.
Beurteilung der DDS durch Frequenzmultiplikation:

\+ F_{in} muß nur ca. so groß wie F_{out} sein

\+ linearer Zusammenhang F_{in} und F_{out}

\+ wenige Stufen trotz hoher Auflösung

– – maximaler Phasenfehler ca. $\frac{T_{\text{in,I}}}{2}$

Bei diesem Verfahren ist es *nicht* möglich, durch Erhöhung der Eingangsfrequenz eine Verringerung des Phasenfehlers zu erreichen, da der wesentliche Beitrag zum Fehler durch die letzte Stufe entsteht. Eine Verbesserung des Phasenfehlers ergibt sich erst durch einen nachgeschalteten Frequenzteiler [Nie93].

2.2.4 Abschließender Vergleich der DDS–Verfahren

In einem abschließenden Vergleich sollen die DDS–Verfahren anhand der im Beispiel 2.1 genannten Anforderungen auf eine PGA–Implementierung untersucht hin werden.

Für den *Akkumulator–Synthesizer* lesen wir aus Bild 2.2.3(a) eine minimal nötige Eingangsfrequenz von $F_{\text{in}} = 250$ MHz ab. Bei 100 Hz Frequenzauflösung wird ein $\lceil F_{\text{in}}/\text{D}F \rceil = 21$ Bit Akkumulator benötigt.
Für den *Teiler–Synthesizer* können die Angaben dem Beispiel 2.1 entnommen werden.

Eine direkte Verwendung der *multiplikativen Frequenzerzeugung* ist nicht möglich, da der auftretende Frequenzfehler sehr groß wäre. Wird die multiplikative Frequenzerzeugung ausgangsseitig durch einen Teiler ergänzt, so soll der günstigste Fall angenommen werden, daß dann am Ausgang des Teilers ein Frequenz– bzw. Phasenfehler auftritt, der genauso groß ist wie der eines Akkumulator–Synthesizers gleicher Eingangsfrequenz. Für die Frequenzauflösung von 100 Hz wird ein siebenstufiger Multiplizierer und zur Erzeugung des geforderten Ausgangsfrequenzbereichs ein (nicht ladbarer) 13 Bit Teiler benötigt.

Für die drei DDS–Verfahren zeigt die Tabelle 2.2.2 die geforderten Taktfrequenzen, den Realisierungsaufwand in CLB bei Verwendung eines Xilinx XC3000 und den Leistungsverbrauch. Während der Realisierungsaufwand in CLB akzeptabel ist, so kann die geforderte Taktfrequenz mit den 1994 verfügbaren Bausteinen noch nicht erreicht werden. Ein noch größerer Nachteil ist hingegen der für einen portable Realisierung nicht akzeptable Leistungsverbrauch aller DDS–Verfahren größer 1 Watt. Die im nächsten Abschnitt vorgestellten hybride Verfahren sind bezüglich des Leistungsverbrauchs wesentlich günstiger.

Typ	F_{in}	CLB–Anzahl	Pin Watt
Akku	0,25 GHz	21	1,375
Teiler	2,25 GHz	16	2,25
Multiplikation		$7 \cdot 3 + 13 \cdot 0,5$	
mit Teiler	0,25 GHz	=27,5	1,718

Tabelle 2.2.2 Vergleich des Realisierungsaufwandes der DDS–Verfahren in PGA–Technik.

2.3 Hybride Verfahren zur Frequenzsynthese

Die hybriden Verfahren benutzen eine Kombination aus analogen und digitalen Prinzipien der Frequenzsynthese. Die hybriden Verfahren sind in der praktischen Realisie-

rung sehr flexible und dennoch recht aufwandsgünstige Verfahren und es existiert deshalb vielfältige Literatur zu diesem Thema [EB91], [PO88], [She90], [Bes87]. Meist kommt hierbei eine Phase–Locked Loop (PLL) zum Einsatz. Der hybride PLL–Synthesizer mit Quarz–Referenz und Teiler in der Rückführung besitzt geringeres Phasenrauschen, hingegen bei feiner Frequenzauflösung und damit großen Teilerwerten in der Rückkopplung neigen diese PLLs zu Instabilitäten [PO88]. Daher ist ein vorrangiges Ziel weiterer Modifikationen des PLL–Synthesizers (Akkumulator als PLL Referenz oder fraktionale M–Synthese), den Teiler in der Rückführung zu verkleinern. Zum Abschluß dieses Abschnitts über hybride Synthesizer wird ein neuartiges Akkumulator–Verfahren vorgestellt, das ohne PLL auskommt.

2.3.1 Die PLL zur Frequenzsynthese

Die klassische Frequenzerzeugung mit einer PLL ist in Bild 2.3.1 skizziert.

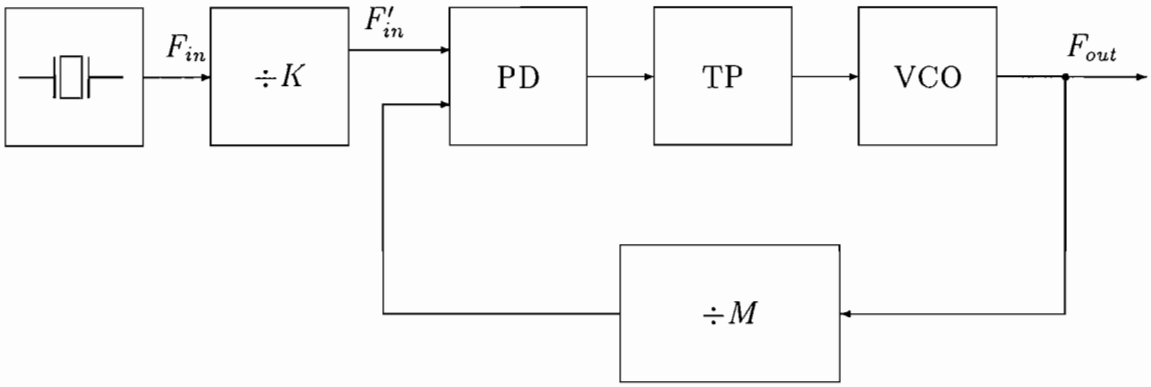


Bild 2.3.1 Digital einstellbare Frequenzerzeugung mit PLL.

Aus einer quarzgenauen Frequenz wird durch Teilung mit K (siehe Abschnitt 2.2.2) eine neue Frequenz F'_{in} erzeugt. Diese Frequenz gelangt auf eine PLL, die in der Rückführung durch einen zweiten Teiler ergänzt ist. Der Phasenregelkreis regelt die beiden Eingangssignale des Phasendetektors so aus, daß deren Phasendifferenz Null wird. Daher gilt am Eingang der PLL

$$F'_{\text{in}} = \frac{F_{\text{in}}}{K} = \frac{F_{\text{out}}}{M} \tag{2.3.1}$$

$$\rightarrow F_{\text{out}} = \frac{M}{K} F_{\text{in}} \tag{2.3.2}$$

Die Frequenzauflösung der PLL ist somit gegeben durch

$$\mathrm{D}F = \left(\frac{M+1}{K} - \frac{M}{K}\right) F_{\text{in}} = \frac{1}{K} F_{\text{in}} = F'_{\text{in}} \tag{2.3.3}$$

Dabei sollte die Zeitkonstante der PLL möglichst groß dimensioniert werden, damit die Ausgangsfrequenz stabil ist. Eine große Zeitkonstante bedeutet aber andererseits, daß bei

einem Umschalten der Frequenzwerte M oder K die PLL lange benötigt (bis zu Minuten), bis sich die neue Ausgangsfrequenz einstellt.

Obwohl die Eingangsfrequenz der PLL $F'_{\text{in}} = F_{\text{in}}/K$ quarzgenau ist, tritt aufgrund von Eigenrauschen der Bausteine und Übersprechen von Störsignalen ein Regelverhalten der PLL und damit verbundenes Phasenrauschen auf. Eine kleine Störspannung am VCO–Eingang führt bei großem Durchstimmbereich bereits zu einer starken Frequenzverschiebung des VCOs. Eine Möglichkeit, das Phasenrauschen möglichst gering zu halten, ist, M klein zu halten. Dies widerspricht der Forderung, M groß zu wählen, um einen möglichst weiten Frequenzbereich durchstimmen zu können. Soll wie im Beispiel 2.1 bei 150 kHz eine Auflösung von 100 Hz erreicht werden, so benötigt man einen Teiler $M = 1500$ in der Rückführung der PLL, die mit einem so großen Teiler nur äußerst schwierig zu stabilisieren ist. Zusammenfassend ergeben sich für die Auswahl der PLL–Parameter die folgenden, teils widersprüchlichen Anforderungen:

- Zeitkonstante der PLL groß $\rightarrow$ Ausgangsfrequenz stabil
- Zeitkonstante klein $\rightarrow$ schneller Frequenzwechsel möglich
- Phasenrauschen klein $\rightarrow$ M klein
- hohe Frequenzauflösung $\rightarrow$ M und K groß

Die Vor– und Nachteile seien folgendermaßen zusammengefaßt:

\+ lineare Frequenzeinstellung über M

\+ Leistungsverbrauch und Komplexität der Schaltung gut

– hohe Frequenzauflösung schwierig

– lange Einschwingzeit

– Phasenrauschen

Es sei nochmals betont, daß der vorgestellte hybride PLL–Synthesizer mit Quarz–Referenz und *kleinem* Teiler in der Rückführung geringeres Phasenrauschen besitzt, hingegen bei feiner Frequenzauflösung und damit großen Teilerwerten in der Rückkopplung neigen diese PLLs zu Instabilitäten [PO88]. Hier können die im nächsten Abschnitt beschriebenen PLL–Synthesizer Verbesserungen bringen, da mit diesem Verfahren wesentlich kleinere Teilerwerte in der PLL–Rückführung realisiert werden können.

2.3.2 Modifizierte PLL–Synthesizer

In den nächsten zwei Abschnitten werden modifizierte PLL–Synthesizer vorgestellt. So kann eine Modifikation des Teilers in der Rückkopplung (Fraktionale M–Synthese) Verbesserungen bezüglich der PLL–Stabilität bringen [Her91b], [EB91], [BL92]. Sehr beliebt ist auch eine Kombinationen aus PLL und Akkumulatortechnik [Her91b], [LSS88], [NS91], [HM88], die ebenfalls eine wesentliche Verringerung des Teilers in der PLL–Rückführung zuläßt.

2.3.2.1 Akkumulator–Synthesizer als PLL–Referenz

Die Forderung bei der DDS mit dem Akkumulator–Synthesizer (siehe Abschnitt 2.2.1) nach möglichst geringer Ausgangsfrequenz um ein geringes Phasenrauschen zu erzielen, läßt sich durch eine anschließende Frequenzvervielfachung mit Hilfe einer PLL erreichen. Die anschließende Frequenzvervielfachung durch eine PLL stellt einen guten Kompromiß dar, wenn eine relativ hohe Ausgangsfrequenz in einem weiten Bereich durchsteuerbar sein soll. Eine entsprechende Schaltung zeigt Bild 2.3.2, wobei ein N Bit breiter Akkumulator zur Anwendung kommt.

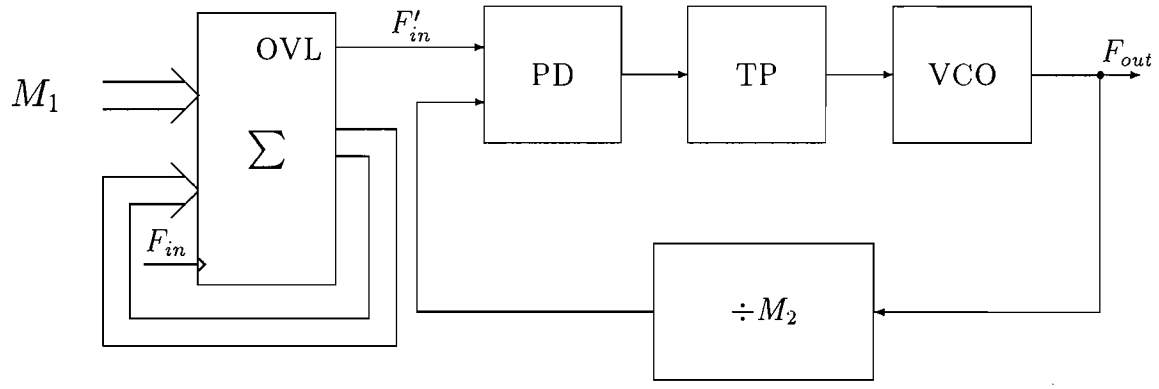


Bild 2.3.2 Kombination von PLL mit Akkumulator.

In diesem Fall muß *nicht* die Frequenz F'_{in}, wie bei der PLL aus Abschnitt 2.3.1, mit der Frequenzauflösung identisch sein. Für die Ausgangsfrequenz der Schaltung ergibt sich

$$F_{\text{out}} = M_2 F'_{\text{in}} \tag{2.3.4}$$

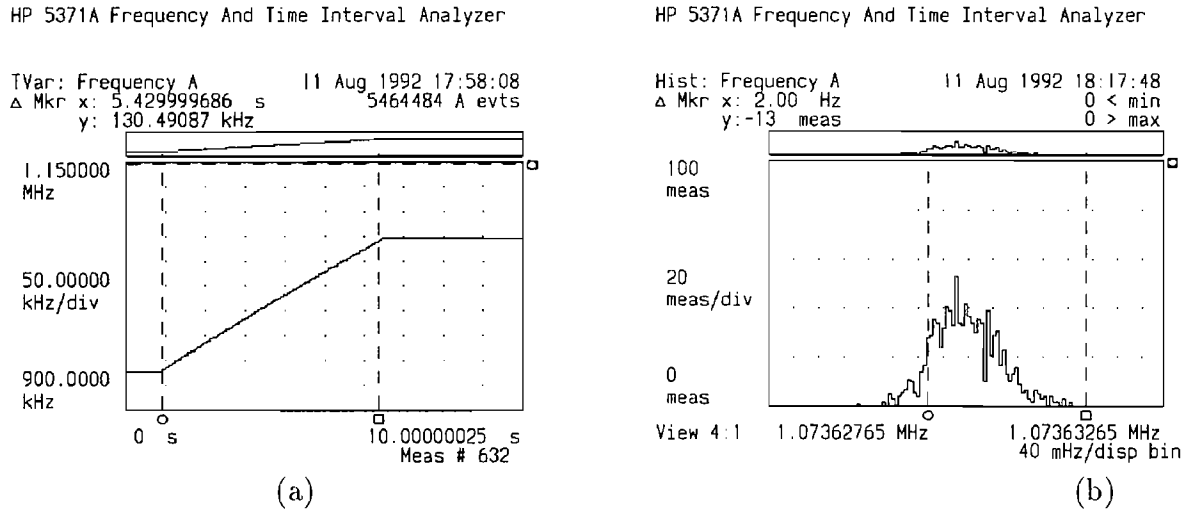


Bild 2.3.3 PLL–Synthesizers mit Akkumulatorreferenz. (a) Einschwingverhalten. (b) Frequenzfehlerverhalten.

$$= \frac{M_1 \cdot M_2}{2^N} F_{\text{in}} \tag{2.3.5}$$

M_2 wird üblicherweise fest und möglichst klein gewählt. Die hohe Frequenzauflösung wird durch eine große Akkumulatorbreite erreicht. Für die Frequenzauflösung folgt mit Gleichung (2.2.2)

$$\mathrm{D}F = \left(\frac{M_1 + 1}{2^N} - \frac{M_1}{2^N}\right) M_2 F_{\text{in}} \tag{2.3.6}$$

$$= \frac{M_2}{2^N} F_{\text{in}} \tag{2.3.7}$$

Die Realisierungen dieses Synthesizers in [AW91], [BB92] und [RS93] zeigen, daß sich dieses Prinzip bei geringem Realisierungsaufwand durch günstiges Frequenzfehlerverhalten und einen geringen Leistungsverbrauch auszeichnet. Beispielhaft sollen die Realisierungsaspekte und zugehörige Meßkurven anhand des Synthesizers aus [AW91] erläutert werden. Bei F_{in}=28 MHz, einem 24 Bit Akkumulator (24 CLB) und Teiler mit $M_2 = 29$ (8 CLB) ergibt sich eine maximale Ausgangsfrequenz von 3,172 MHz und eine Frequenzauflösung von 1,669 Hz. Das Bild 2.3.3(a) zeigt das Verhalten des Synthesizers beim Umschalten des Empfängers von Senders DCF37 nach Sender DCF54. Die Abtastfrequenz liegen um den Faktor 8 über denen in Abschnitt 1.2 genannten Signalfrequenz. Bild 2.3.3(b) zeigt den zugehörigen Frequenzfehler des Synthesizers von ca. 2 Hz. Der Hardware–Aufwand bleibt mit 32 CLBs und einem zusätzlichen PLL–Baustein recht gering. Der Leistungsverbrauch eines ähnlichen Aufbaus mit F_{in}=6 MHz und 21 Bit Akku zur Mischfrequenzerzeugung in [BB92] benötigt 31 mW und liegt damit um einen Faktor 40 besser als die DDS–Synthesizer gleicher Güte, siehe Tabelle 2.2.2.

Zusammenfassend läßt sich über die Eigenschaften des Synthesizers mit Akkumulator als PLL–Referenz sagen, daß gegenüber der PLL–Struktur sich auf Kosten der Komplexität eine Verbesserung des Einschwingverhaltens ergibt. Die Stabilität der PLL und das Phasenrauschen werden durch kleine Werte von M_2 verbessert, der Akkumulator fügt aber weiteres Phasenrauschen hinzu, so daß bezüglich Phasenrauschen gegenüber der konventionellen PLL aus Abschnitt 2.3.1 keine wesentliche Verbesserung zu erwarten ist.

Eine andere Modifikation der Schaltung aus Bild 2.3.2 führt auf ein Verfahren nach [HM88] bei dem der Akkumulator zur Erzeugung von „Phasenzwischenzuständen“ benutzt wird. Dadurch kann eine weitere Reduktion der Einschwingdauer und des Phasenrauschens erreicht werden. Eine genauere Analyse dieser Schaltung ist in [HM88] zu finden.

2.3.2.2 Fraktionale M–Synthese

Eine weitere Möglichkeit die kritische Größe des Teilers in der Rückführung der PLL zu verringern, stellt die **fraktionale M–Synthese** dar.

Bei der fraktionalen M–Synthese wird eine modifizierte PLL–Struktur benutzt, wobei der Teiler in der Rückführung der PLL um $+1/0^{\dagger}$ variiert wird. Somit erreicht man, daß

† Es sind auch Variationen um 0 ± 1 oder $0 \pm 1 \pm 2$ denkbar, die jedoch keinen Vorteil gegenüber der $M + 1/M$–Fragment bringen.

die Eingangsfrequenz der PLL auch die Frequenzauflösung vorgibt, da gemittelt quasi jede Frequenz zwischen $M \cdot F_{\text{in}} \leq F_{\text{out}} \leq (M+1)F_{\text{in}}$ erzeugt werden kann.

Da die PLL relativ schmalbandig dimensioniert wird, erfolgt kein abrupter Übergang der Ausgangsfrequenz an den Umschaltpunkten des Teilers, sondern ein kontinuierlicher Übergang, so daß die PLL das beim Umschalten entstandene Phasenrauschen verringert. Dennoch verbleibt gegenüber der ursprünglichen Schaltung aus Bild 2.3.1 eine *erhebliche* Verschlechterung bezüglich des Phasenrauschens. Durch Summation des Phasenfehlers mittels eines Akkumulators kann eine Verringerung des Phasenrauschens erreicht werden [Bes87, S. 97].

Im Akkumulator wird der fraktionale Wert (z. B. 0,3) des Teilungsverhältnisses aufaddiert und der Überlauf im Akkumulator führt zur Auslöschung eines Impulses in der PLL–Rückführung. Der D/A gewandelte Wert des Akkumulators dient dabei zum Ausgleich des beim „Auslöschen" eines Impulses entstehenden Sprungs im Phasenfehler. Dieses Vorgehen ist nochmals in Bild 2.3.4 veranschaulicht.

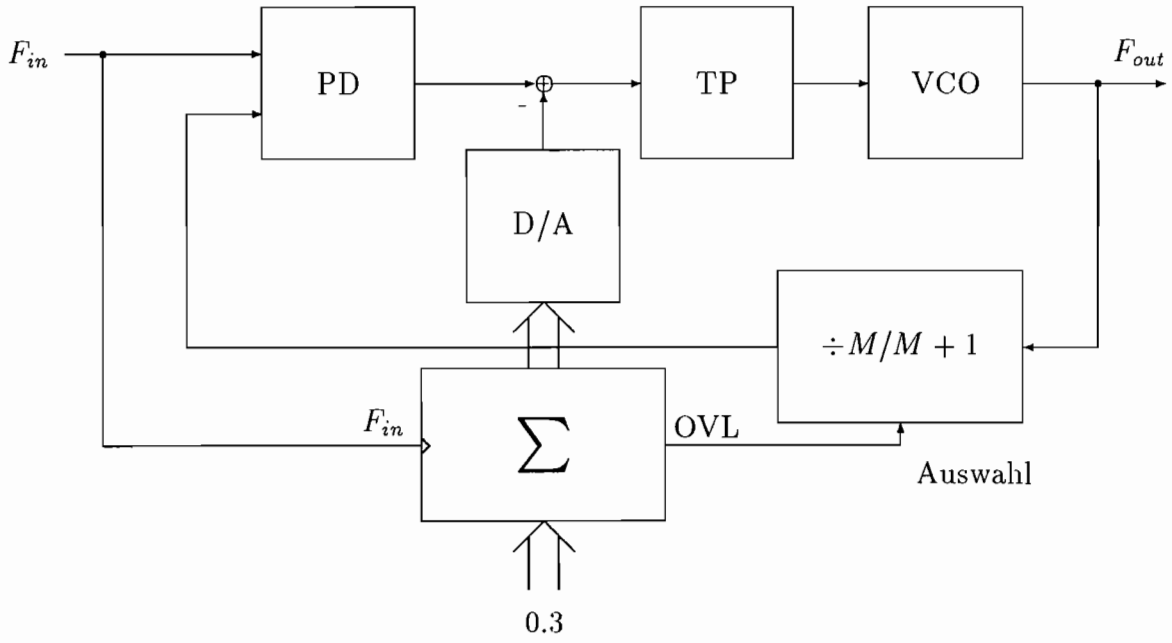


Bild 2.3.4 Fraktionale M–Synthese mit Phasenfehlerkompensation durch Akkumulator.

In [Her91b] wird vorgeschlagen, die Akkumulation über 2 Stufen durchzuführen. Dies führt zu einem verstärkten Umschalten des Teilers zwischen M und $M+1$ und bewirkt eine Verringerung des Phasenrauschens der PLL. Setzt man mehr als 2 Akkumulatoren ein, so bedeutet dies, daß nun als Teilungsverhältnis auch $M-1$ und $M+2$ realisiert werden muß.

Zusammenfassend läßt sich festhalten, daß durch die fraktionale Synthese zum einen ein erhebliches Phasenrauschen hinzugefügt wird zum anderen die kritische Größe des Teilers in der Rückführung erheblich verringert werden kann. Mit den in [Her91b] vorgeschlagenen Modifikationen ist eine wesentliche Reduktion des Frequenz– bzw. Phasenfeh-

lers möglich und die Eigenschaften fallen ähnlich günstig aus wie die Eigenschaften von Akkumulator–Synthesizer mit PLL.

2.3.3 Neuer hybrider Frequenzsynthesizer mit Phasen– und Frequenzfehler Kompensation durch Akkumulator–Rest–Division

Zum vollständigen Verständnis des neuen Verfahrens ist es nötig, nochmals das Phasen– und Frequenzverhalten des Akkumulatorsynthesizer genauer zu analysieren.

2.3.3.1 Verallgemeinerung

Im folgenden sind Verallgemeinerungen zum Verhalten des Akkumulator–Synthesizer mit Beweisen angeben, die bereits in den Bildern 2.2.1 und 2.2.2 veranschaulicht sind und Voraussetzung für das Verständnis des neuen Verfahrens sind.

I: Eine Periode ist $\lceil \frac{2^N}{M} \rceil T_{\text{in}}$ oder $\lfloor \frac{2^N}{M} \rfloor T_{\text{in}}$ lang.

II: In einem Zyklus, bei dem der Akkumulator von Null ausgehend wieder den Wert Null besitzt, enthält exakt $\frac{M}{ggT(M,2^N)}$† Überläufe des Akkumulators.

Als Einzelfälle sind zu nennen:

- M ungerade $\rightarrow$ ggT$(M, 2^N) = 1$, d. h. ein Zyklus besitzt M Perioden.
- $M = 2^m \rightarrow M/\text{ggT}(M, 2^N) = M/2^m = 1$, d. h. ein Zyklus besitzt exakt eine Periode.
- $M = K \cdot 2^i$ mit ggT$(K, 2)$=1 $\rightarrow M/\text{ggT}(M, 2^i) = M/2^i = K$, d. h. ein Zyklus besitzt K Perioden.

III: Der Phasenfehler ist stets positiv und der maximal auftretende Phasenfehler beträgt

$$\delta\phi_{max} = \widehat{T}(i) - i \cdot \widetilde{T}\Big|_{\text{max}} = \left(1 - \frac{1}{M}\right) T_{\text{in}} \tag{2.3.8}$$

IV: Der Phasenfehler ist gleichverteilt über das Intervall $[0 \cdots \delta\phi_{max}]$

V: Der Akkumulator–Rest ist direkt proportional zum Phasenfehler:

$$\boxed{\delta\phi(i) = \frac{Rest(i)}{M} T_{\text{in}}} \tag{2.3.9}$$

Im folgenden werden diese Verallgemeinerungen bewiesen.

† $ggT(M, 2^N)$ bedeutet hierbei: größter gemeinsamer Teiler von M und 2^N.

zu I Der Akkumulator läuft über, wenn seine Summe größer 2^N wird. In jeder Periode wird M addiert, wobei $\lfloor\frac{2^N}{M}\rfloor M < 2^N$ gilt.‡ Für den anfänglichen Rest in jedem Zyklus gilt $0 \leq$Rest$\leq M-1$. Nun ist es möglich, daß : Rest$+\lfloor\frac{2^N}{M}\rfloor M < 2^N$ gilt (z. B. für Rest=0). Ebenfalls ist : Rest$+\lfloor\frac{2^N}{M}\rfloor M \geq 2^N$ möglich. Im zweiten Fall erfolgt dann sofort die Modulo–Operation und die Periodendauer ergibt sich zu $\lfloor\frac{2^N}{M}\rfloor T_{\text{in}}$, während im ersten Fall eine zusätzliche Addition nötig ist, d. h. $\lfloor\frac{2^N}{M}\rfloor + 1 = \lceil\frac{2^N}{M}\rceil$ und somit eine Periodendauer von $\lceil\frac{2^N}{M}\rceil T_{\text{in}}$ entsteht.

zu II Der Akkumulator führt immer eine Operation modulo 2^N aus, da die Bit–Breite des Akkumulators N Bit beträgt. Der Rest beim Akkumulatorüberlauf muß kleiner M sein, da vor der letzten Addition: Akkumulator $< 2^N$ galt und nun M modulo 2^N addiert wird. Somit können maximal M verschiedene Überläufe stattfinden, da der Rest ein Zustandsregister darstellt, das maximal M Zustände besitzt $(0 \cdots M-1)$.

Enthält die binäre Darstellung von M andererseits in den niederwertigsten Bit i Nullen (vor der ersten Eins), so werden von den möglichen M Zuständen nur $M/2^i$ Zustände eingenommen, da bei der Modulo–2^N–Operation nie die (2^i-1) untersten Zuständen eingenommen werden können. Diese anschauliche Erklärung ist in [MB92b, Anhang **B**] formal bewiesen und wurde in Simulationen für 1000 verschiedene Werte M überprüft und bestätigt.

zu III Das exakte Periodenende entsteht dort, wo die Gerade aus den Akkumulatorwerten den Maximalwert des Akkus schneidet, siehe Bild 2.2.1(b). Der Überlauf kann aber nur nach der nächsten Akkumulation detektiert werden. Der größte Rest und damit Fehler ergibt sich zu $M-1$. Da der Fehler immer kleiner T_{in} bleibt, folgt Fehler kleiner $\frac{M-1}{M}T_{\text{in}}$.

zu IV Ist M ungerade so werden alle Zustände des Restes erreicht und die Gleichverteilung stellt sich ein. Ist $M = K \cdot 2^i$, so treten (siehe **II**)) jeweils die unteren (2^i-1) Zustände des Restes nicht auf. Da aber die restlichen K Zustände $(0, 2^i, 2\cdot 2^i \cdots (K-1)2^i)$ alle auftreten und äquidistant sind, stellt sich wieder eine Gleichverteilung ein.

zu V Aus **III** und **IV** folgt, daß die Differenz zwischen Detektion des Überlaufs und dem Schnittpunkt zwischen maximalen Akkumulatorinhalt und der linearen Extrapolation der Akkumulatorwerten am Ende jeder Addition, somit der Phasenfehler, unmittelbar proportional zum Rest des Akkumulators ist. Der Proportionalität Faktor ergibt sich wie bei **III** zu T_{in}/M.

Die Regeln **I–IV** verkürzen die Analyse von Frequenz– und Phasenfehler wesentlich, da eine vollständige Statistik bereits nach Simulation von einem Zyklus L erstellt werden kann; während Regel **V** unmittelbar als Vorschrift für eine Fehlerkompensation eingesetzt werden kann.

‡ Der triviale Fall $M = 2^m$, d. h. die Zyklusdauer beträgt genau eine Periode, sei nicht betrachtet.

2.3.3.2 Hybride Struktur zur Phasenfehlerkompensation

Führt man die Divisionsvorschrift entsprechend Gleichung (2.3.9) analog oder digital aus, so kann eine Kompensation des Frequenz– und Phasenfehlers (siehe die Bilder 2.2.2(a) und (b)) mit geringem Aufwand erreicht werden. In der Folge werden einige günstige Realisierungen beschrieben.

2.3.3.3 Hardware–Konzepte

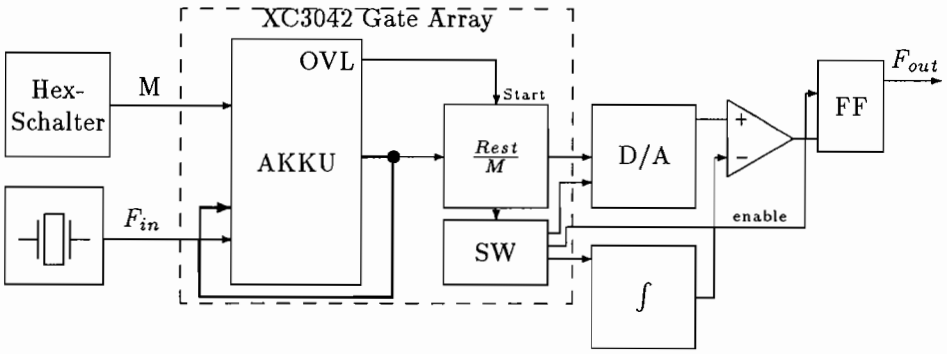


Bild 2.3.5 Blockschaltbild der Testschaltung.

Bei einer analogen Lösung würde die Division analog ausgeführt und die erreichbare Verbesserung hängt wesentlich von der Linearität des Dividierers ab, wobei sicherlich nur recht schwierig Genauigkeiten besser als 1% erreicht werden können. Da mit D/A–Wandlern in 8 Bit Breite bereits bessere Ergebnisse (Fehler$< 0,4\%$) zu erwarten sind und diese Bausteine preiswert sind, soll im folgenden nur die digitale Lösung weiterbetrachtet werden. Die digitale Lösung ist in Bild 2.3.5 skizziert. Akkumulator und Division können hierbei günstig in einem PGA implementiert werden, wobei die Division mittels Table–Lookup, seriell, iterativ oder mit einem CORDIC–Zustandsmaschine realisiert werden kann, siehe z. B. [Spa76, S. 107-146], [Wal71] oder [MB91]. Eine Beurteilung der verschiedenen Prinzipien ist in Tabelle 2.3.1 gegeben. Die Schaltung aus Bild 2.3.5 sei anhand des Timings erläutert: Bei einem Akkumulatorüberlauf erfolgt die Übernahme des Restes in das Dividend–Register. Anschließend wird die Division ausgeführt und der Quotient schließlich D/A gewandelt. Das analoge Signal ist dann direkt proportional der zu realisierenden Voreilung der Taktflanke. D. h.

$$\text{DA-Wert} = 0\text{V} \rightarrow \delta\phi = 0 \cdot T_{\text{in}} \tag{2.3.10}$$

$$\text{DA-Wert} = 5\text{V} \rightarrow \delta\phi = T_{\text{in}} \tag{2.3.11}$$

Führt man nun diesen D/A–Wert und eine negative Rampe einem Komparator zu, so erfolgt bei Gleichheit eine positive Flanke am Ausgang. Diese Flanke kann direkt ausgewertet werden (z. B. als S&H–Flanke). Wird während des Vergleichszeitraums ein

TFF freigegeben, so erfolgt beim Umschalten des Komparators ein Wechsel des TFFs. Theoretisch ist es auch möglich, Rampe und Komparator digital zu realisieren. Allerdings ist der Realisierungsaufwand für die volldigitale Lösung wesentlich *größer* als bei der hybriden. Dies sei an einem Zahlenbeispiel verdeutlicht: Die Akkumulatorfrequenz betrage 50 MHz und es soll Phasenfehlerkompensation mit einem 8 Bit D/A–Wandler durchgeführt werden. Für die Rampe und den Komparator ergibt sich hier eine Taktfrequenz von $2^8 \cdot 50$ MHz=12,8 GHz, die eventuell noch in diskreter Logik realisierbar ist, aber nicht als verlustleistungsarme PGA–Schaltung. Der analoge Integrator benötigt hier lediglich eine „slew rate“ von 5V/20ns=250V/μs.

Division und D/A–Wandlung mit Komparation nach Bild 2.3.5 können sich zeitlich überlappen (Pipelining), da dies lediglich zu einer konstanten Verzögerung führt und eine längere Wandlungszeit ermöglicht. Für die Division und für D/A–Wandlung mit Komparation steht jeweils eine vollständige Periode der Ausgangsfrequenz zur Verfügung. Bereitet die Divisionszeit Probleme, so kann die Division selbst als Pipeline–Struktur (z. B. als CORDIC–Pipeline, [MB91]) realisiert werden.
Bei der Division (Akkumulatorrest$\div M$) bestehen drei verschiedene Möglichkeiten den Quotienten zu runden:

a) Runden zur *nächsten* Ganzzahl (round).

b) Runden zur nächst *größeren* Ganzzahl (ceil).

c) Runden zur nächst *kleineren* Ganzzahl (floor).

Wobei als Verfahren mit dem geringsten Aufwand das Verfahren **c)**, d. h. Vernachlässigen des Restes, zu nennen ist. In Bild 2.3.6 werden die drei Verfahren bezüglich Phasen– und Frequenzfehler bei konstanter Akkumulatorfrequenz (F_{in}) und variabler Ausgangsfrequenz (F_{out}) miteinander verglichen. Bei Runden zur *nächsten* Ganzzahl kann beim Phasenfehler bis zu einem Bit D/A–Breite gewonnen werden, während sich beim Frequenzfehler keine Verbesserung ergibt, da hier ein konstanter Phasenfehler, wie er bei „ceil“ und „floor“ vorliegt, zu keiner Verschlechterung führt. Die Werte des Phasenfehlers liegen hierbei geringfügig oberhalb denen des Frequenzfehlers.

Eine weitere Variante des Synthesizers wäre eine direkte Umsetzung von Gleichung (2.3.9) in bis zu $(M-1)$ Verzögerungsleitungen mit jeweils $\tau = T_{\text{in}}/M$ Verzögerung

	Table–Look–Up	Seriell	Iterativ	CORDIC
Zykluszeit (100 MHz FPGA)	100 ns	125 ns	>100 ns	165 ns
Zyklenzahl	1	n bis 2n+2	~n	2n
Max. Bit–Breite	8 Bit	>16 Bit	>16 Bit	>16 Bit
Rechenzeit für 10 Bit	100 ns für 8 Bit	~1 ms	~1 ms	~2 ms
Schaltungsaufwand	Externes EPROM	gering	hoch	mittel
Integrierbarkeit (für XILINX–FPGA)	schlecht	gut	schlecht	noch gut
Max. Akkutakt	10 MHz	8 MHz	10 MHz	6 MHz
Max. Ausgangsfrequenz	10 MHz	1 MHz	1 MHz	300 kHz

Tabelle 2.3.1 Vergleich der Division–Prinzipien mit FPGAs.

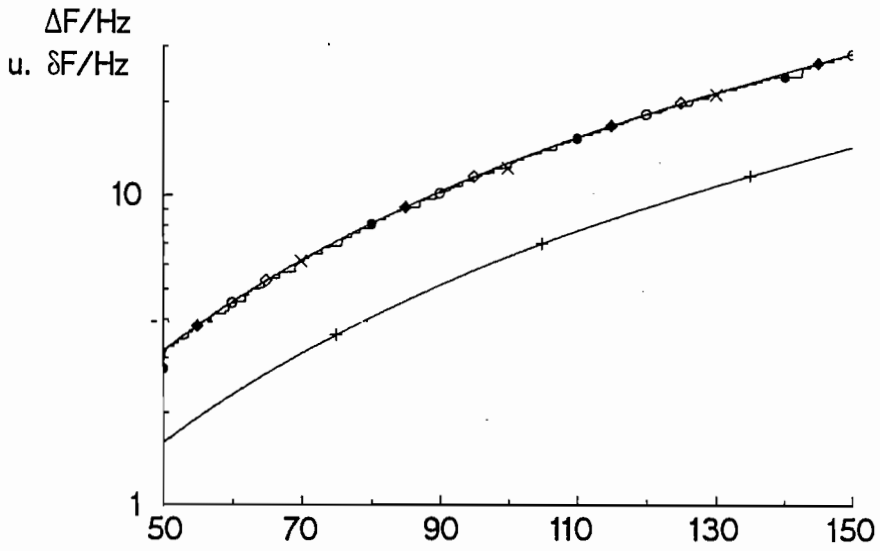


Bild 2.3.6 Phasenfehler und Frequenzfehler für verschiedene M bei festem F_{in}: (+) Phasenfehler bei round. (◆) Phasenfehler bei ceil. (◇) Phasenfehler bei floor. (×) Frequenzfehler bei round. (•) Frequenzfehler bei ceil. (○) Frequenzfehler bei floor.
(N=17; $F_{\text{out}} = 50 \cdots 150$ kHz ($M = 500 \cdots 1500$);
DF=100 Hz; F_{in}=26,2144 MHz)

[LSS88]. Für verschiedene Werte von M benötigt man allerdings auch verschiedene Verzögerungsleitungen mit entsprechenden Verzögerungselementen τ. Sollen z. B. 32 verschiedene Frequenzen erzeugt werden, so benötigt man für eine vollständige Fehlerkompensation bereits $\sum_{i=1}^{32} i - 1 = 216$ verschiedene Verzögerungselemente. Der Aufwand steigt somit quadratisch mit M.

Abschließend sei noch eine vereinfachte Variante vorgeschlagen, die eingesetzt werden könnte, wenn die digitale Division nicht realisiert werden kann. Unter Einsatz von einem zusätzlichen multiplizierenden D/A–Wandler (MDAC) kann die Division durch M in eine entsprechende Multiplikation mit $1/M$ umgesetzt werden [Wie92a]. Hierbei sollte die Auflösung des MDAC–Wandlers wesentlich über dem des Wandlers für die D/A–Wandlung des Restes liegen, da durch den zweiten Wandler zusätzliches Quantisierungsrauschen entsteht. Für diesen Fall (Auflösung Wandler MDAC $1/M \gg$ Auflösung Akku–Rest–Wandler) ergeben sich dann im wesentlichen die gleichen Resultate für Frequenz– und Phasenfehler wie bei der Lösung mit digitaler Division. Wird der Akku–Rest unmittelbar gewandelt, so resultiert bei einem D/A–Wandler mit üblicherweise 2^i Stufen, daß die obersten Worte des D/A–Wandlers nicht benutzt werden. Dies führt zu einem Verlust von bis zu einem Bit an Auflösung, für die Realisierung des Synthesizers mit MDAC–Division.

2.3.3.4 Beschreibung des Testaufbaus

Um dieses neue Verfahren in der Praxis zu erproben, wurde eine Testschaltung [Wol92] (siehe Bild 2.3.5) mit folgenden Daten aufgebaut:

- Ausgangsfrequenz 50 kHz - 600 kHz.
- Frequenzauflösung 100 Hz.
- Realisierung des digitalen Teils in einem XILINX Gate Array (XC3042).

In der Testschaltung wurde ein 18 Bit breiter Akkumulator verwendet. Der Dividierer wurde als 16 Bit breiter iterativer Dividierer, der ein 8 Bit breites Ergebnis liefert, realisiert. Die Ablaufsteuerung sorgt für die zeitlich korrekte Abfolge von Überlauf, Division und Phasenverschiebung.

Der so ermittelte Phasenfehlerwert muß dann noch in eine Zeitverschiebung umsetzt werden. Der digitale Ausgangswert des Gate Arrays wird mit einem 8 Bit Digital/Analog–Wandler in eine proportionale Spannung umgesetzt.

Die Realisierung des Rampensignals mit einem als Integrator betriebenen Operationsverstärker erwies sich im Versuchsaufbau als nicht ausreichend linear und war anfällig für Offset–Fehler. So wurde als Alternative eine Schaltung mit einem als steuerbare Stromquelle betriebenen OTA (Operational Transductance Amplifier) gewählt. Mit dem erzeugten Strom wird ein Kondensator aufgeladen, an dem so eine linear mit der Zeit ansteigende Spannung ansteht.

2.3.3.5 Meßergebnisse

In der Abbildungen 2.3.7 ist die Häufigkeit des Auftretens einer bestimmten Frequenz über der dazugehörenden Frequenzabweichung aufgetragen. Im unkorrigierten Fall (Bild

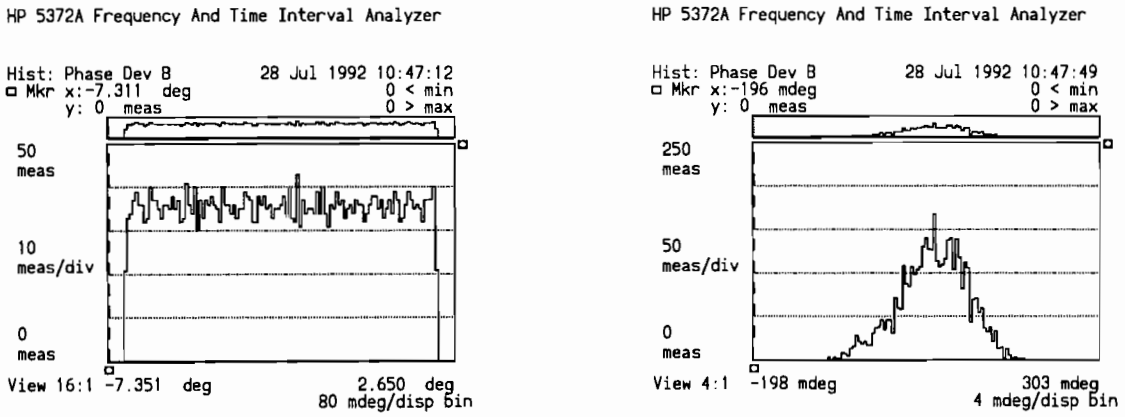
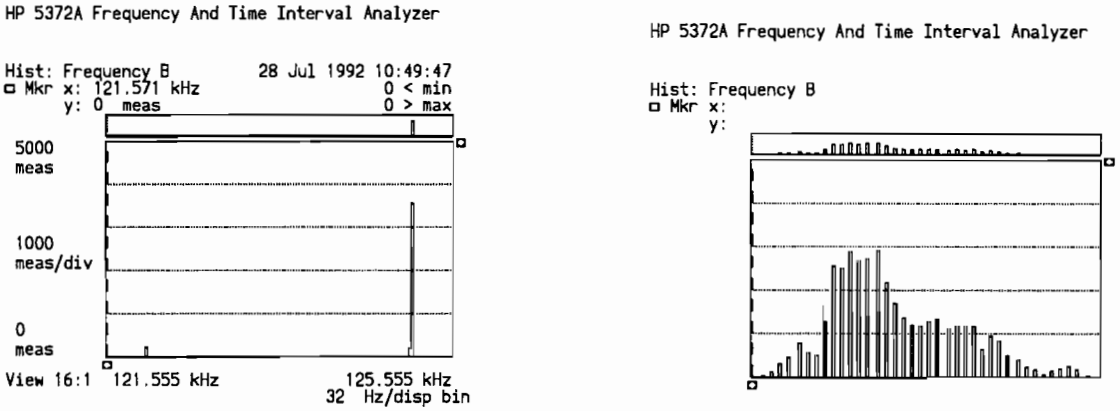


Bild 2.3.7 (a) Verteilung des Phasenfehlers ohne Korrektur (b) Verteilung des Phasenfehlers mit 8 Bit Korrektur.



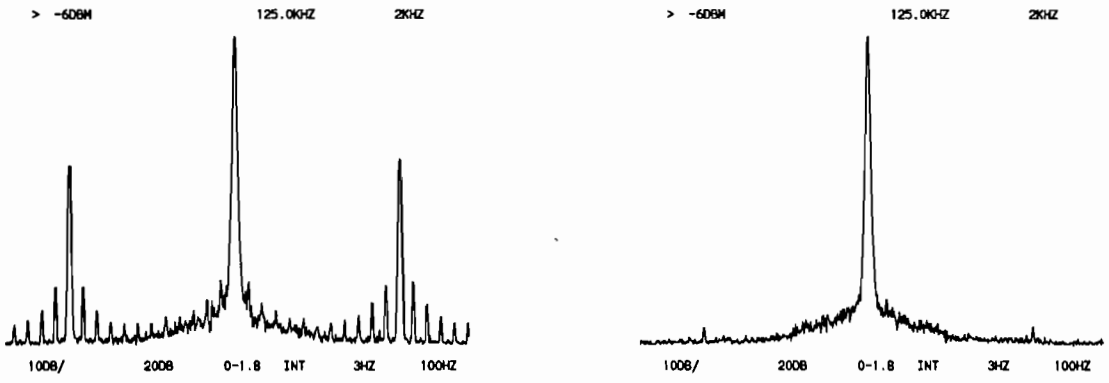
(a) (b)

Bild 2.3.8 (a) Verteilung der Frequenzen ohne Korrektur. (b) Verteilung der Frequenzen mit 8 Bit Korrektur. X–Achse von 124,85 kHz bis 124,95 kHz.

2.3.7(a)) kann man gut die von der Theorie vorausgesagte Gleichverteilung der Phasenfehler erkennen.

In der Abbildung 2.3.8 ist dasselbe für die auftretenden Frequenzen durchgeführt. Im unkorrigierten Fall (Bild 2.3.8(a)) sieht man deutlich die beiden diskreten Frequenzen, aus denen sich der DDS–Output zusammensetzt.

In den letzten beiden Messgraphen (Bild 2.3.9) sind die Ergebnisse der Messung mit einen gewöhnlichen Spektralanalysator dargestellt. Da die Einschwingzeit der Filter des



(a) (b)

Bild 2.3.9 (a) Ausgangsspektrum des Synthesizers ohne Korrektur. (b) Ausgangsspektrum des Synthesizers mit 8 Bit Korrektur.

Analysators zu groß ist, erhält man nicht wie in Abbildung 2.3.8(a) die zwei echten Frequenzen sondern eine mittlere Frequenz mit Seitenbändern. Auch in diesen Plots ist die Verbesserung der spektralen Reinheit klar erkennbar.

2.4 Zusammenfassung und Einordnung des neuen Verfahrens

In Abbildung 2.4.1 sind die Meßergebnisse des vorangehenden Abschnitts den theoretischen Ergebnissen gegenübergestellt. Bis zu einer Wandlerbreite von 4 Bit folgt die gemessene Kurve gut der Theorie. Die Abflachung der Kurve bei höheren Bit–Breiten ist durch Nichtlinearitäten und Störeinflüsse in der Experimentierschaltung bedingt.

Die Ergebnisse zeigen, daß die Kompensation eine Reduktion des Phasenrauschens um einen Faktor von ca. 60-80 erbringt. Schon um ein diesem einfachen Testaufbau adäquates Ergebnis zu erbringen, müßte ein einfacher Akkumulatorsynthesizer mit einer Taktfrequenz von über 300 MHz betrieben werden. Bei Verwendung der heute erhältlichen schnelleren Gate–Arrays und einem Aufbau in „Hybrider Schaltungstechnik“ sollte es ohne weiteres möglich sein, die äquivalente Akkumulatorfrequenz auf 1 Gigahertz zu steigern [MBW94b]. Die Schaltung ist mit 96 CLB Aufwand bei einer Verlustleistung von 0,47 W (optimiert: 0,04 W) bzw. 1,065 W (optimiert: 0,1 W) für Digitalteil bzw. Analogteil durchaus für eine Mixed–Signal–Asic Implementierung interessant [MBW94a].

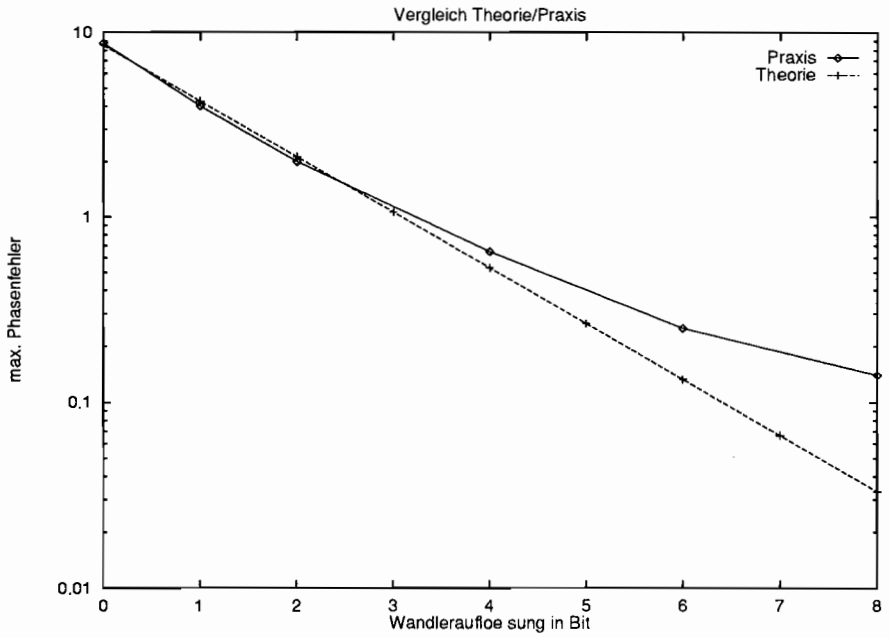


Bild 2.4.1 Vergleich des gemessenen mit dem berechneten Phasenfehler.

Zum Abschluß soll das neue Verfahren mit den verschiedenen Prinzipien zur Frequenzerzeugung bezüglich Verlustleistung und Genauigkeit verglichen werden. Die Genauigkeit ist hierbei als „UND“–Verknüpfung von Stabilität (Frequenz/Phasenfehler), Frequenzauflösung und Einschwingdauer des Synthesizers zu verstehen. Das Bild 2.4.2 dient dabei

als grobe Klassifikation der Verfahren, da Genauigkeit und Leistungsverbrauch gegeneinander austauschbar sind und somit die Verfahren „Ortskurven“ in der Ebene bilden. Gezeigt sind typische Einsatzbereich der verschiedenen Verfahren.

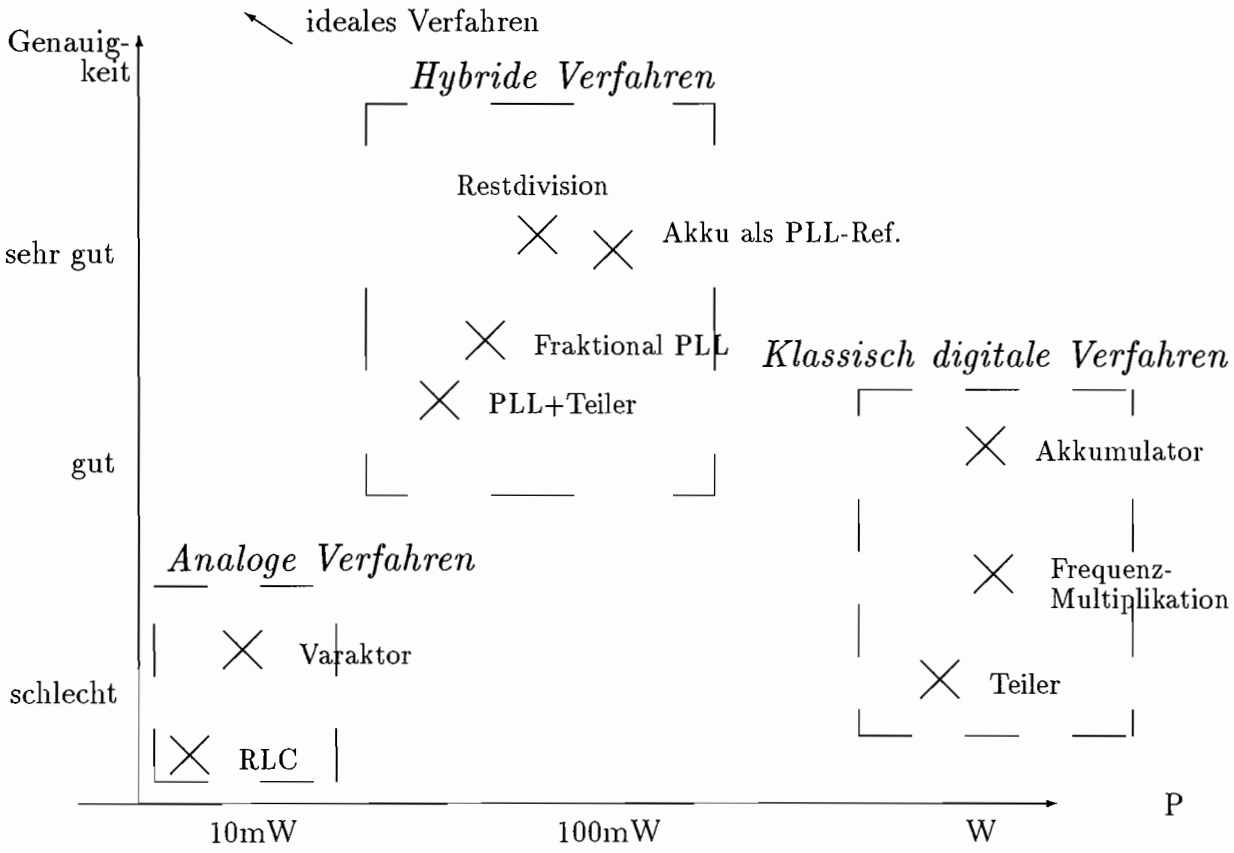


Bild 2.4.2 Vergleich von verschiedenen Frequenzsynthesizern.

Abschließend bleibt zu sagen, daß das hier vorgestellte neue hybride Verfahren einen gelungenen Kompromiß zwischen den Vorteilen einer voll digitalen Lösung und der geringen Verlustleistung der analogen Verfahren darstellt. Im Vergleich zu anderen hybriden Verfahren ergeben sich bei geringem Aufwand ebenso gute oder teilweise bessere Eigenschaften bis hin zu hohen Frequenzen.

Kapitel 3

Selektion und A/D–Wandlung für Bandpaßsignale

Bei einem digitalen Empfänger sollten Selektion (Filterung) und A/D–Wandlung ganzheitlich betrachtet werden, da aus der Wahl des Wandlers (Nyquistwandler, Unterabtaster oder Bandpaßwandler) die Anforderungen für die notwendige Filterung erwachsen. Im folgenden werden einige Alternativen für die A/D–Wandlung von schmalbandigen Bandpaßsignalen aufgezeigt, wobei das Bandpaß $\Sigma\Delta$–Prinzip sich als besonders vorteilhaft erweisen wird. In den folgenden Abschnitten wird dann ein Algorithmus für den Entwurf des Bandpaßwandlers vorgestellt, der auf die besonderen Anforderungen, die für einen universellen digitalen Empfänger entstehen, Rücksicht nimmt.

Die Schnittstelle zwischen Analog– und Digitalwelt, und somit die A/D–Wandler, ist immer wieder Thema vieler Veröffentlichungen. In den Artikeln werden sowohl Bereiche neuer wissenschaftlicher Forschungsergebnisse (z. B. [AS83], [Can85] oder [SS90]) erörtert als auch anwendungsorientiertere Aspekte wie Wandler–Prinzip, Auflösung, Umsetzrate und Leistungsverbrauch (z. B. [Goo91], [Pic91], [E+92], [M+91] und [P+92b]) im kommerziellen Bereich behandelt. Im Bereich der neuen Produkte und Forschungsergebnisse stehen hier häufig $\Sigma\Delta$–A/D–Wandler im Vordergrund, siehe z. B. [P+91].

Zunächst sollen mögliche Alternativen für die A/D–Wandlung eines Bandpaßsignals erörtert werden.

3.1 A/D–Wandlung für Bandpaßsignale

Oft tritt in der Signalverarbeitung das Problem auf, daß ein Frequenzgemisch/Multiplex vorliegt, aber nur ein schmaler Bereich† des Spektrums das interessierende Nutzsignal enthält. Ziel der Untersuchung sei es nun im folgenden einen A/D–Wandler zu entwerfen, der im Bereich 50-150 kHz mit variabler Mitten– und Bandbreitenfrequenz Bandpaß–Signale digitalisiert, um sie einer digitalen Auswertung bzw. Demodulation zugänglich zu machen. Das gesuchte System sei entsprechend Bild 3.1.1 graphisch veranschaulicht.

† Z. B. LW–Antennen Signal, AM–ZF–Signal, RDS–Basisband oder Telefonsignale im Frequenzmultiplex.

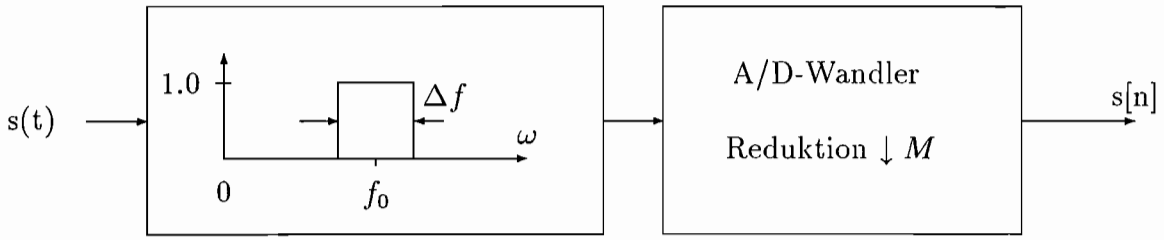


Bild 3.1.1 Ideales System zur Wandlung von Bandpaßsignale in ein Digitalsignal.

Um eine hohe Reduktion durchführen zu können, muß das Filter eine möglichst *hohe Ordnung* besitzen. Dabei sind mehrere Prinzipien denkbar:

a) Schmalbandige Filterung hoher Ordnung mit anschließender Unterabtastung (Integer–Band–Sampling)

b) Abtastung mit Nyquist–Rate ‡ mit anschließender digitaler Filterung

c) Mischung und schmalbandige Filterung

d) Der BP–ΣΔ–A/D–Wandler

zu a) Eine schmalbandige Filterung hoher Ordnung bis 500 kHz mit rein passiven (L/C) oder mit aktiven OP–Schaltungen ist nur für eine feste Frequenz möglich. Alternativ sind durchstimmbare S/C–Filter nur für Frequenzen von 50-150 kHz bei Ordnungen von 2-4 erhältlich. Um die geforderte Filterordnung zu erreichen, müßten sehr viele Filter in Reihe geschaltet werden. Durch die Schalter in den S/C–Filtern sind außerdem bestenfalls 60-70 dB Signal/Rauschabstand zu erwarten.

zu b) Bei Nyquist–Abtastung und digitaler Filterung eines schmalbandigen Bandpaß-Signals zeigt sich schnell, siehe z. B. [MB91, S. 7], daß der schnellste, leistungsfähigste DSP die Filter hoher Ordnung nicht realisieren kann. Eine Parallelisierung der Filterung mittels „polyphase–Strukturen“ in mehrere DSPs ist möglich [CR83, S. 79-81], erscheint aber wegen des hohen Aufwands und Leistungsverbrauchs nicht geeignet.

zu c) Die Mischung beinhaltet natürlich die typischen analogen Probleme wie Temperaturstabilität, Kreuzmodulation, Bauteiltoleranz und Langzeitstabilität. Für Bandbreiten $\Delta f < 10$ kHz stellt es ein beliebtes klassisches Prinzip dar ([Bra90], [RS93]), da für diesen Fall Keramik/Quarz–Filter eingesetzt werden können und so eine hohe Ordnung der Filterung möglich wird.

‡ Unter *Nyquistfrequenz* versteht man hierbei die maximale Signalfrequenz bei der es unter Abtastung mit *Nyquistrate* zu keiner Verletzung des Shannonschen (1949) Abtasttheorems bzw. des Abtasttheorems nach H. Raabe, der diese Beziehung schon 1939 [Lü78] ableitete, kommt.

zu d) Die Vorteile der ΣΔ–Technik, basierend auf den geringen Anforderungen an die Genauigkeit des Analogteils und dem hohen Signal/Rauschverhältnis am Ausgang des digitalen Filters, werden ausgenutzt und die ΣΔ–Technik auf Bandpaßsignale hin ausgelegt.

Der Entwurf eines Bandpaßwandlers in ΣΔ–Technologie erscheint, vor allem für eine spätere monolithische Lösung zur digitalen Verarbeitung von Bandpaßsignale, die interessanteste Lösung zu sein. Daher werden im folgenden mögliche Realisierungsformen des A/D–Wandlers besprochen.

Bevor in Abschnitt 3.3 der ΣΔ–A/D–Wandler für Bandpaßsignale untersucht wird und im Abschnitt 3.3.1 mögliche Realisierungen vorgestellt werden, erfolgt in Abschnitt 3.2 eine kurze Erläuterung zu den zwei wichtigsten Entwurfsmerkmalen „Noise Shaping“ und „Überabtastung“ am Beispiel des konventionellen ΣΔ–A/D–Wandlers, d. h. eines A/D–Wandlers für Tiefpaßsignale.

3.2 Der konventionelle ΣΔ–A/D–Wandler

ΣΔ–A/D–Wandler zeichnen sich durch eine ideale Kombination eines analogen Eingangskreises und eines digitalen Ausgangskreises aus. Ideal deshalb, da an den analogen Eingangkreis geringe Anforderungen bezüglich Bauteilgenauigkeit, Komplexität der Schaltung und Temperaturstabilität gestellt werden und die hohe Auflösung (z. Z. typisch 16 bis 20 Bit) durch ein digitales Ausgangsfilter (Dezimator) erreicht wird, siehe Bild 3.2.1. Durch die Fortschritte in der Mixed–ASIC–Entwicklung gelang es erstmals 1989 der Firma Analog Device, einen monolithischen ΣΔ–A/D–Wandler auf den Markt zu bringen.

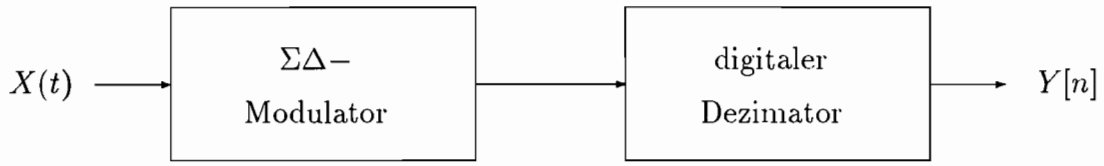


Bild 3.2.1 Prinzipieller Aufbau eines ΣΔ–A/D–Wandlers.

Konventionelle ΣΔ–A/D–Wandler besitzen einen analogen Eingangskreis, der auch als ΣΔ–Modulator bezeichnet wird. In Bild 3.2.2(a) ist das Eingangsfilter mit der anschließenden Schwellwertbildung (A/D) dargestellt. Dieser Schwellwertbildner ist ein 1 Bit A/D–Wandler, weshalb die ΣΔ–A/D–Wandler auch als 1 Bit Wandler bezeichnet werden. Hinter dem Schwellwertbildner entsteht durch einen entsprechenden Takt ein serieller Datenstrom von typisch 1 bis 10 MBit. Dieser serielle Datenstrom gelangt dann auf einen rein digitalen Dezimator (Tiefpaß mit anschließender Abtastratenreduktion), so daß am Ausgang des Wandlers ein 16 bis 20 Bit breites Datenwort bei einer Rate von bis zu 150 kHz entsteht.

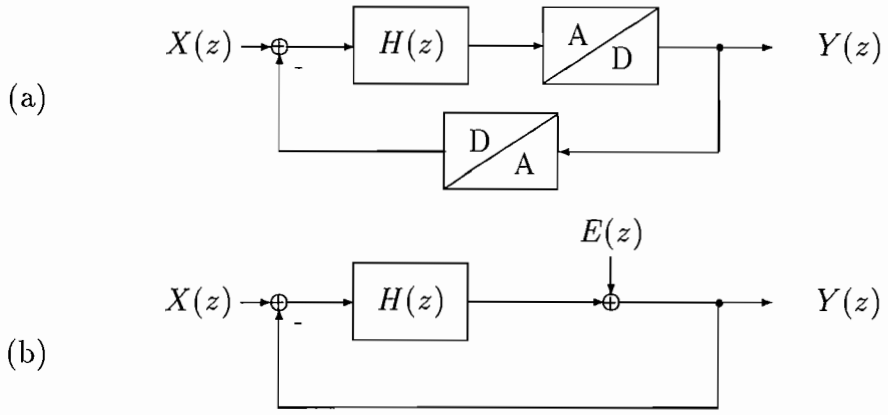


Bild 3.2.2 (a) Blockschaltbild eines ΣΔ–Modulators. (b) Kleinsignalnäherung im z–Bereich.

3.2.1 Das Prinzip des Noise Shaping

Der analoge Eingangskreis stellt für das Nutzsignal ein Anti–Aliasing–Filter dar. Das Quantisierungsrauschen, das durch den Schwellwertbildner entsteht, wird durch die richtige Dimensionierung von $H(z)$ so gefiltert, daß es für die im Nutzsignal enthaltenen Frequenzen möglichst klein wird (Noise Shaping). Dies führt zu einer Verstärkung der Rauschleistung bei höheren Frequenzen, was aber wegen der anschließenden Dezimation zu keiner Degradation im Nutzsignal führt.

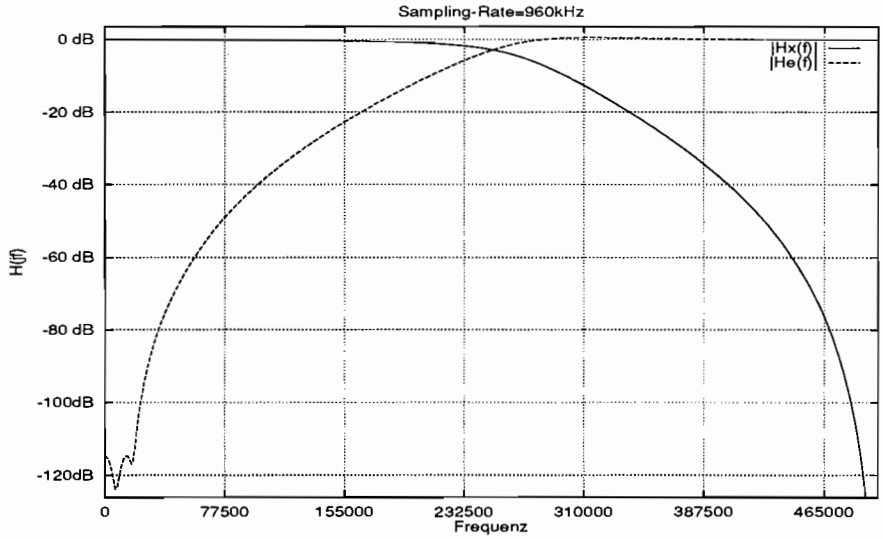


Bild 3.2.3 Übertragungsfunktionen für das Nutzsignal ($H_x(f)$) und für das Quantisierungsrauschen ($H_e(f)$).

Der Begriff ΣΔ–Modulator erklärt sich daraus, daß im Gegensatz zum ΔΣ–Modulator zuerst eine Integration und dann eine Differenzbildung durchgeführt wird.

Die Signalübertragungsfunktion (STF) ergibt sich, wenn die Verzögerung durch die D/A–Wandlung vernachlässigt wird, entsprechend Bild 3.2.2(b) zu

$$\text{STF}: \qquad H_x(z) \;=\; \frac{H(z)}{1+H(z)} \tag{3.2.1}$$

und für die Rauschübertragungsfunktion (NTF) ergibt sich:

$$\text{NTF}: \qquad H_e(z) \;=\; \frac{1}{1+H(z)} \tag{3.2.2}$$

Die Abhängigkeit von NTF und STF ergibt sich zu

$$H_x(z) \;=\; 1 - H_e(z) \tag{3.2.3}$$

Somit kann von der Charakteristik der Rauschübertragungsfunktion $H_e(z)$ direkt auf die Signalübertragungsfunktion $H_x(z)$ geschlossen werden, so daß gewünschte Eigenschaften von $H_e(z)$ beim Entwurf von $H_x(z)$ berücksichtigt werden können.

Wählt man als Schleifenfilter $H(z)$ einen einfachen Integrator $H(z) = 1/(1-z^{-1})$, so spricht man von einem $\Sigma\Delta$–A/D–Wandler *erster* Ordnung. Dabei ist die Ordnung des Gesamtsystems aus Bild 3.2.2(b) auf Grund des Rückkopplungszweiges um einen Grad höher als die Filterordnung des Schleifenfilters $H(z)$.

Für einen Wandler 2–ter Ordnung wird meist der $\Sigma\Delta$–A/D–Wandler erster Ordnung durch einen Allpaß ergänzt, siehe z. B. [BM87]. Beim $\Sigma\Delta$–A/D–Wandler N–ter Ordnung wird im allgemeinen eine Filterübertragungsfunktion der Form

$$H(z) \;=\; \frac{a_0 + a_1 z + a_2 z^2 + \cdot a_N z^N}{b_0 + b_1 z + b_2 z^2 + \cdots + b_N z^N} \tag{3.2.4}$$

verwendet. Der Entwurf eines Wandlers N–ter Ordnung wird in Abschnitt 3.3 erörtert.

3.2.2 S/N–Gewinn durch Dezimation

Die Überabtastung in Verbindung *mit* dem „Noise Shaping“ führt bei Dezimation zu einem erheblichen S/N–Gewinn gegenüber der üblichen Überabtastung mit Reduktion. Zur Veranschaulichung des für $\Sigma\Delta$–A/D–Wandler sehr bedeutsamen Sachverhalts werden der Dual–Slope–Wandler [Hil89, S. 360], die konventionelle Abtastung oberhalb der Nyquistrate mit Dezimation und ein $\Sigma\Delta$–A/D–Wandler bei Abtastratenreduktion gegenübergestellt. Dem Nutzsignal sei ein weißes gaußverteiltes Rauschen überlagert.

Für das **Dual–Slope–Prinzip** ist sofort einsichtig, daß es bei einer Verdoppelung der Schrittfrequenz zu einer Halbierung der kleinsten Stufenhöhe kommt und damit zu einem Bit Gewinn:

$$N(2f) \;=\; \frac{1}{4}N(f) \;\;\widehat{=}\;\; 6 \text{ dB oder 1 Bit Gewinn} \tag{3.2.5}$$

Der Gewinn bei **Überabtastung mit Dezimation** wird anschaulich, wenn man eine Abtastung eines Sinus–Signals bei Vielfachen der Periodendauer betrachtet: Für die

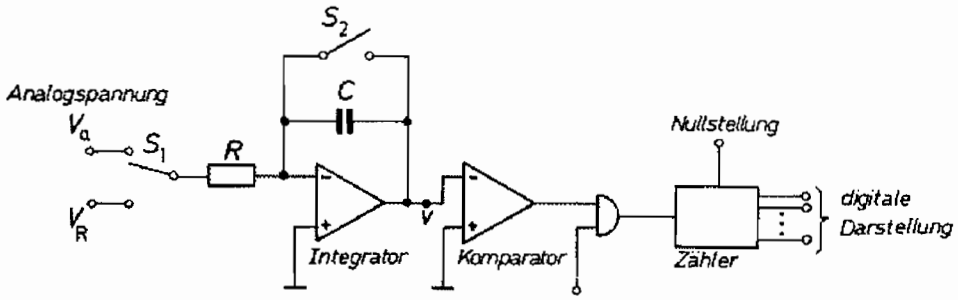


Bild 3.2.4 Der Dual–Slope–Wandler nach [Hil89, S. 360].

Zusammenfassung bzw. Summe von M mittelwertfreien ($\bar{n} = 0$) gaußverteilten Rauschsignalen, die statistisch unabhängig seien, ergibt sich entsprechend [Pap86, S. 111+S. 178] die Rauschleistung mit Hilfe der Erwartungswertbildung zu

$$N = \mathcal{E}\left\{\sum_{i=1}^{M}(\mathbf{n}_i - \bar{n})^2\right\} = \mathcal{E}\left\{\sum_{i=1}^{M}\mathbf{n}_i^2\right\} = M \cdot \mathcal{E}\left\{\mathbf{n}_i^2\right\} = M \cdot N_1 \qquad (3.2.6)$$

N_1 bzw. S_1 sei hierbei die Rausch– bzw. Signalleistung des einzelnen Abtastwerts. Für die Summe von M gleichen Abtastwerten eines mittelwertfreien Signals ($\bar{y} = 0$) ergibt sich folgende Signalleistung

$$S = \mathcal{E}\left\{\left(\sum_{i=1}^{M}\mathbf{y}_i - M\bar{y}\right)^2\right\} = \mathcal{E}\left\{(M\mathbf{y}_i - M\bar{y})^2\right\} = \mathcal{E}\left\{M^2\mathbf{y}_i^2\right\} = M^2 \cdot S_1 \qquad (3.2.7)$$

Für den Quotienten aus Signal– und Rauschleistung bei einer Reduktion um M Abtastwerte ergibt sich daher:

$$\frac{S}{N} = M\frac{S_1}{N_1} \qquad (3.2.8)$$

Das bedeutet für eine Reduktion um den Faktor $M = 2$ einen Gewinn von 2, d. h. 3 dB.

Für den **ΣΔ–A/D–Wandler** ist die Berechnung des S/N–Gewinns etwas komplexer. Sie soll anhand des ΣΔ–A/D–Wandlers erster Ordnung veranschaulicht werden. Die Herleitung der gesuchten Werte wird durch eine Abbildung der Schaltung aus Bild 3.2.2(a) in den z–Bereich und durch die Berechnung der Rauschleistung im Nutzband mit „Noise Shaping“ erreicht. Für das Ausgangssignal des A/D–Wandlers entsprechend Bild 3.2.2(b) erhält man folgende Summe aus Nutzsignal– und Störsignalkomponenten:

$$Y(z) = H_x(z)X(z) + H_e(z)E(z) \tag{3.2.9}$$

Für den Integrator erster Ordnung wird der Ansatz $I_1(z) \approx 1/(1-z^{-1})$ gewählt.

$$H_{e1}(z) = 1/(1+I_1(z)) \approx 1 - z^{-1} \tag{3.2.10}$$

Damit ergibt sich für die Rauschleistungsdichte $S_{ee}(f)$ nach der Bewertung durch das Filter

$$S_{ee}(f)/\sigma_e^2 = \left|H_{e1}(z)H_{e1}(z^{-1})\right|_{z=e^{j\omega/f_a}} \tag{3.2.11}$$

$$= \left|4\sin^2(\pi f/f_a)\right| \tag{3.2.12}$$

Liegt die Abtastfrequenz wesentlich oberhalb der Signalfrequenz ($f_0 \ll f_a$), so folgt für die Rauschleistung σ_N^2 im Bereich $[0, f_0]$

$$\sigma_N^2 = 2/f_a \int_0^{f_0} 4\sigma_e^2 \sin^2(\pi f/f_a)\mathrm{d}f \tag{3.2.13}$$

$$\approx \frac{8}{3}\pi^2\sigma_e^2(f_0/f_a)^3 \tag{3.2.14}$$

Eine Verdoppelung der Abtastfrequenz f_a bewirkt somit eine Verringerung der Rauschleistung um einen Faktor 8, was einem Gewinn von 9 dB entspricht.

Für den ΣΔ–A/D–Wandler zweiter Ordnung ergeben die entsprechenden theoretischen Betrachtungen in [AS83, S. 363] einen Gewinn von 15 dB bei Verdoppelung der Abtastfrequenz. Auf die Herleitung für Wandler höherer Ordnung soll hier verzichtet werden, da im Kapitel über die Bandpaß–Wandler darauf näher eingegangen wird. Die graphische Erklärung wird anhand von Bild 3.2.3 deutlich. Hier erkennt man, daß das Rauschsignal im Nutzband durch die Filterung („Noise Shaping“) wesentlich verringert wurde.

Die Ergebnisse dieses Abschnitts sind in Tabelle 3.2.1 zusammengefaßt. Deutlich ist die Überlegenheit des ΣΔ–Verfahrens zu erkennen.

Wandlungsprinzip	Gewinn bei $f_a' = 2 \cdot f_a$
Nyquistabtastung mit Dezimation	3 dB
Dual–Slope–Wandler	6 dB
ΣΔ–A/D–Wandler 1. Ordnung	9 dB
ΣΔ–A/D–Wandler 2. Ordnung	15 dB

Tabelle 3.2.1 Vergleich des S/N–Gewinns für verschiedene Dezimierungsarten bei Verdoppelung der Taktfrequenz.

3.2.3 Noise Shaping für Bandpaßsignale

Im folgenden wird der ΣΔ–Ansatz für Bandpaßsignale präzisiert:

- Der analoge Eingangskreis führt ein „Noise Shaping“ für den Bandpaßbereich des Nutzsignals durch, d. h. die Übertragungsfunktion für das Quantisierungsrauschen stellt eine Bandsperre dar, siehe Bild 3.2.5. D. h. *Nullstellen* der Rauschübertragungsfunktion sind im Signalband zu wählen, während dort keine Pole auftreten dürfen.

- Die Filterordnung des Wandlers sollte gerade sein, da für BP–Signale mit reeller Übertragungsfunktion Pole immer konjugiert komplex auftreten und im allgemeinen das Auftreten von Polen bei $f = 0$ und $f = f_a/2$ unnötig ist.

- Für die Übertragungsfunktion des Nutzsignals wäre zu wünschen, daß im Bereich des Nutzsignals eine Bandpaß–Charakteristik entsteht, damit eine ausreichende Dämpfung für die Aliasing–Frequenz $f_a - f_0$ gewährleistet ist.

- Beim Entwurf der *Pole* der Rauschsignalübertragungsfunktion $H_e(z)$ sollte ein Hochpaß bzw. eine Bandsperre entworfen werden, damit eine ausreichende Dämpfung der Signalübertragungsfunktion $H_x(z) = 1 - H_e(z) \cong 0$ bei der Aliasing–Frequenz $f_a - f_0$ erreicht wird.

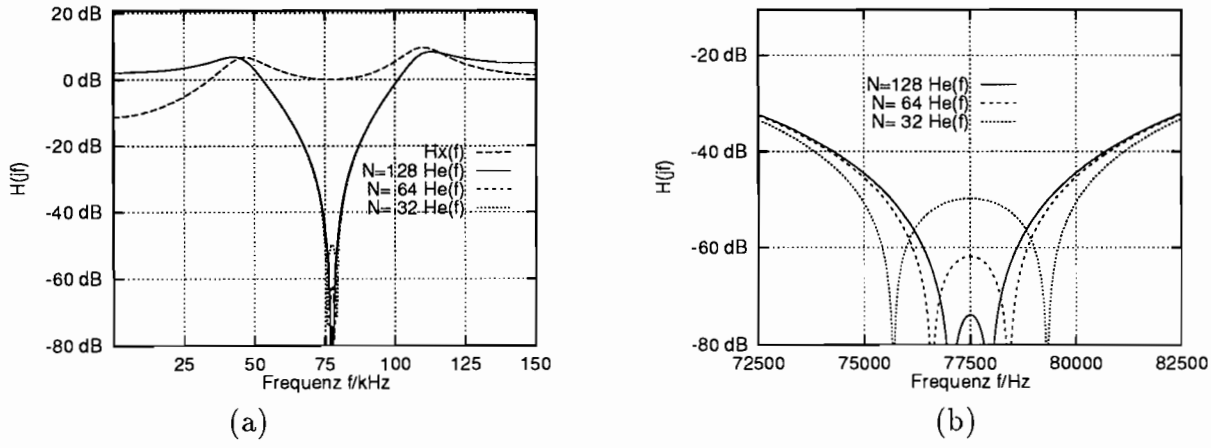


Bild 3.2.5 Signalübertragungsfunktion $H_x(f)$ und Rauschübertragungsfunktion $He128(f)$, $He64(f)$ und $He32(f)$ (Überabtastfaktor $N = f_a/f_b$ =128, 64 bzw. 32 d. h. 3,6; 1,8 bzw. 0.9 kHz Nutzsignal–Bandbreite bei $f_a = 12 \cdot 77{,}5$ kHz) für den BP–ΣΔ–A/D–Wandler. (b) ist eine Ausschnittsvergrößerung der Rauschübertragungsfunktion $H_e(f)$, die für DCF77–Signale optimiert ist.

3.2.4 S/N–Gewinn für den Bandpaßwandler

Für einen qualitativen Vergleich zwischen TP– und BP–ΣΔ–Wandler soll im folgenden betrachtet werden, wie sich eine Verdoppelung der Abtastfrequenz bzw. eine Halbierung

der Signalfrequenz auf das S/N–Verhältnis am Ausgang des Wandlers auswirkt. Angenommen sei, daß ein Bandpaß N–ter Ordnung genau N Nullstellen in der Rauschübertragungsfunktion realisieren kann. Nimmt man weiterhin an, daß die Pole der Gesamtübertragungsfunktion nicht im Signalfrequenzbereich liegen und die Nullstellen $\phi_0 = 2\pi f_0/f_a$ für realisierbare reelle Systeme nur konjugiert komplex auftreten, so kann man für die Rauschsignalübertragungsfunktion bei gerader Ordnung ansetzen:

$$|H_e(z)| = K_0' \prod_{i=1}^{N} |z - z_{0i}| = K_0' \prod_{i=1}^{N/2} |z - z_{0i}| \prod_{i=1}^{N/2} |z - z_{0i}^*| \tag{3.2.15}$$

Für die Rauschleistung im Signalfrequenzband folgt mit $z = e^{j2\pi f/f_a}$:

$$\sigma_N^2 = \int_{f_0-f_b}^{f_0+f_b} |H_e(f)|^2 \mathrm{d}f \tag{3.2.16}$$

$$\sim \int_{f_0-f_b}^{f_0+f_b} \prod_{i=1}^{N} |e^{j2\pi f/f_a} - e^{j2\pi f_0/f_a}| \prod_{i=1}^{N} \underbrace{|z - z_{0i}^*|}_{\text{Konstant bei } f \in f_0 \pm f_b} \mathrm{d}f \tag{3.2.17}$$

mit der Substitution $f' = f - f_0$ und $f_b/f_a \ll 1$ folgt

$$\sigma_N^2 \sim \int_{-f_b}^{f_b} |e^{j2\pi f/f_a} - 1|^N \mathrm{d}f \tag{3.2.18}$$

$$\sim (f_b/f_a)^{N+1} \tag{3.2.19}$$

Damit ergibt sich für eine Verdoppelung der Abtastfrequenz (f_a) beim Wandler 2. bzw. 4. Ordnung eine um den Faktor 8 bzw. 32 geringere Rauschleistung und somit ein Gewinn von 9 bzw. 15 dB. Dies ist weitestgehend unabhängig von dem Nullstellenwinkel, solange die Pole nicht in den Signalbereich fallen.

Dies wird auch anhand der Ergebnisse aus Bild 3.2.6 deutlich. Allerdings ist hier nicht eine Verdoppelung der Abtastfrequenz, sondern eine Halbierung der Signalfrequenz vorgenommen worden. Liegen allerdings breitbandige Signale (gleichförmiges Spektrum) vor, so führt die Halbierung der Signalbandbreite zwangsläufig zu einer Halbierung der Signalleistung und bei Verdoppelung des Überabtastfaktors ergeben sich statt 9 und 15 dB jeweils 6 und 12 dB für den Wandler 2. bzw. 4. Ordnung. Andererseits ist es natürlich nicht sinnvoll, schmaler als die Signalbandbreite zu filtern, so daß die Annahme eines sinusförmigen Eingangssignals und die damit verbundene Peak–S/N–Darstellung insbesondere für LW–Signale und ganz besonders für die sehr schmalbandigen FU–AM–Signale durchaus sinnvoll erscheint.

Der S/N–Gewinn in dB bei Bandbreitenverdoppelung ist für verschiedene Bandbreiten und Ordnung 2 und 4 durch numerische Berechnung mittels Rechner bestimmt worden und in Bild 3.2.6 graphisch dargestellt. Es wurde entsprechend [AS83] ein Quantisierer angenommen, der eine auf das Intervall (-1,1) gleichverteilte Rauschamplitude erzeugt. Da die größte Auflösung des Wandlers im sogenannten „Overload–Bereich" erzielt wird, wurde die Cosinus–Amplitude mit der Quantisierer–Maximalamplitude gleichgesetzt. Damit folgt

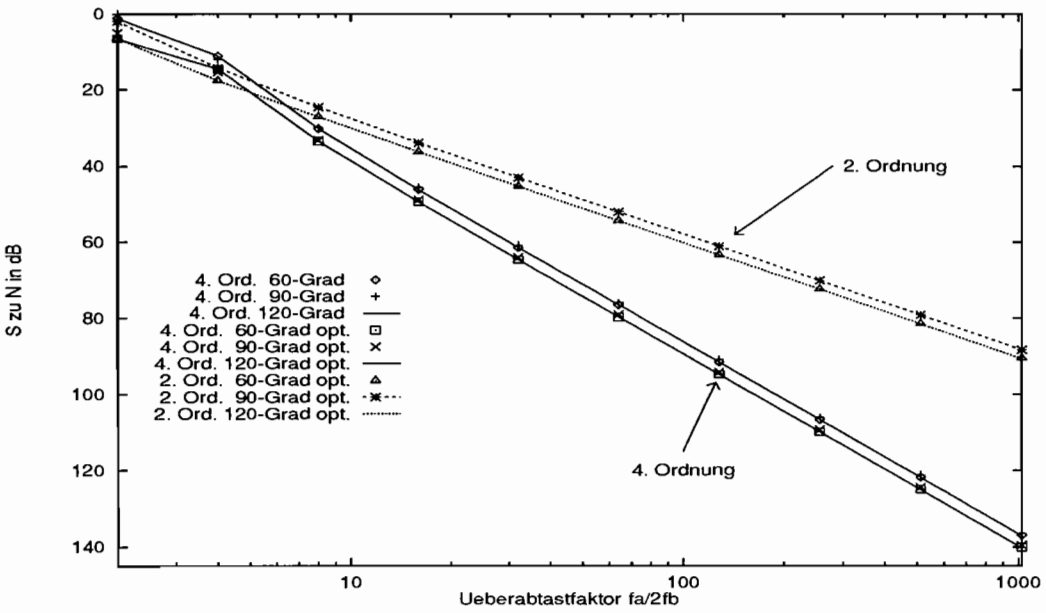


Bild 3.2.6 S/N–Verhältnisse beim BP–ΣΔ–A/D–Wandler 2. und 4. Ordnung bei Halbierung der Signalbandbreite.

für die eingangsseitige Signalleistung $P_s = 0,5$ und für die ungefilterte Rauschleistung $\sigma_e^2 = 1/3$. Bestimmt wurde somit das Peak–S/N–Verhältnis für ein cosinus–förmiges Eingangssignal:

$$S/N_{\mathrm{dB}} = -10\log_{10} N/S \;\; = \;\; -10\log_{10}\sigma_e^2 PF/P_s = -10\log(\frac{2}{3}FP) \qquad (3.2.20)$$

wobei F die Rauschleistungsverstärkung für das Nutzfrequenzband und P die Gesamtrauschleistungsverstärkung darstellen:

$$F = \frac{\int_{f_0-f_b}^{f_0+f_b} S_{\nu\nu}(f)df}{\int_0^{f_a/2} S_{\nu\nu}(f)df} \qquad P = \frac{\int_0^{f_a/2} S_{\nu\nu}(f)df}{\int_0^{f_a/2} S_{ee}(f)df} \qquad (3.2.21)$$

Da die maximale Ausgangsleistung des Quantisierers Eins ist, muß die Summe aus Nutz– und Störkomponenten die Ungleichung $P_s + \sigma_N^2 = P_s + P/3 < 3$ erfüllen. Dieses wichtige Kriterium wird bei der konkreten Parameterbestimmung in Abschnitt 3.3 eine wichtige Rolle spielen.

Plaziert man die Nullstellen von $H_e(z)$ so, daß die entstehende Rauschleistung minimal wird, siehe Bild 3.2.5(b), so erkennen wir aus den Ergebnissen in Bild 3.2.6, daß für den Wandler 4. Ordnung bei gleichem Überabtastverhältnis durch die Optimierung der Nullstellenlage ein Gewinn größer 3 dB zu erwarten ist.

Ein Vergleich der Ergebnisse des TP–ΣΔ–A/D–Wandlers (Gleichung (3.2.14)) mit denen für BP–Wandler (Gleichung (3.2.14)), macht zweierlei deutlich:

- Ein Bandpaß besitzt bei gleicher Flankensteilheit im Vergleich zum TP–Filter die doppelte Ordnung, d. h. Bandpaß 2. Ordnung $\widehat{=}$ TP 1. Ordnung und Bandpaß 4. Ordnung $\widehat{=}$ TP 2. Ordnung.

- Die nutzbare Auflösung des Wandlers ergibt sich nun anders als beim TP-ΣΔ-A/D-Wandler nicht aus dem Verhältnis der Nyquistfrequenz zur Abtastfrequenz, sondern aus dem Verhältnis der Bandbreite des Nutzsignals zur Abtastfrequenz.

Mit dieser Einschränkung sind die Ergebnisse für TP-Wandler aus Tabelle 3.2.1 und [LS87, S. 461] auch auf den BP-Wandler anwendbar. Die Ergebnisse aus Bild 3.2.6 für den nicht optimierten Fall stimmen mit den zwei Beispielen aus [Dre91] für $f_a/f_b = 128$ bzw. 64 überein.

Mögliche Realisierungen des BP-ΣΔ-A/D-Wandlers

Der Entwurf des BP-ΣΔ-A/D-Wandlers beinhaltet zwei Teile: analoge Eingangsstufe und digitales Ausgangsfilter. Zunächst werden mögliche Realisierungen des Analogteils besprochen, dann wird eine spezielle Optimierung des BP-ΣΔ-Entwurf für digitale Abtastempfänger angegeben. Im weiteren wird dann der Entwurf eines effizienten digitalen Bandpaßfilters besprochen.

3.3 Entwurf des Analogteils

Die Anforderungen an den Analogteil bezüglich Genauigkeit und Stabilität sind *nicht* sehr hoch. Lediglich die Durchstimmbarkeit des Filters bereitet Entwicklungsaufwand. Für die Realisierung des Dezimationsfilters existieren eine Vielzahl von Schaltungsmöglichkeiten [Mar82, S. 252-264]. Für eine monolithische Lösung bietet sich, wenn die Auflösungsanforderungen an die zu realisierende Schaltung nicht sehr groß sind, eine Switched/Capacitor (S/C)-Lösung an [SJF92], [T+92]. Eine (S/C)-Lösung bedingt aber lange Entwicklungszeiten und hohe NRE-Kosten und soll deshalb nicht weiter diskutiert werden. Für eine mit diskreten Bauteilen zu realisierende Schaltung sind prinzipiell, wenn die Mittenfrequenz digital einstellbar sein soll, zwei Möglichkeiten zu unterscheiden:

a) Realisierung eines durchstimmbaren Filters mit Hilfe von multiplizierenden D/A-Wandlern oder digital einstellbaren Widerständen (sogenannte E²POT) bei fester Taktfrequenz des 1 Bit Quantisierers

b) Durchstimmbarkeit des Filters über die Taktfrequenz des 1 Bit Quantisierers

Bei der Realisierung nach **a)** benutzt man vorzugsweise OP-Schaltungen mit Mehrfachgegenkopplung. Hierbei können Bandbreite und Bandpaßfrequenz f_0 unabhängig voneinander eingestellt werden, siehe z. B. [TS86, S. 419]. Nachteilig sind hierbei folgende zwei Punkte:

1. Die Auflösung sowohl von D/A-Wandler als auch der E²POTs nimmt bei der Taktfrequenz, wie sie durch den 1 Bit Quantisierer entsteht, oberhalb von 1 MHz stark ab.

2. Die zu realisierenden digitalen Filter benötigen für N verschiedene Signalfrequenzen ebenfalls N verschiedene Koeffizientensätze. Dadurch wird der Einsatz von Filtern mit geringem Aufwand, wie z. B. den Frequenzabtastfiltern, unmöglich.

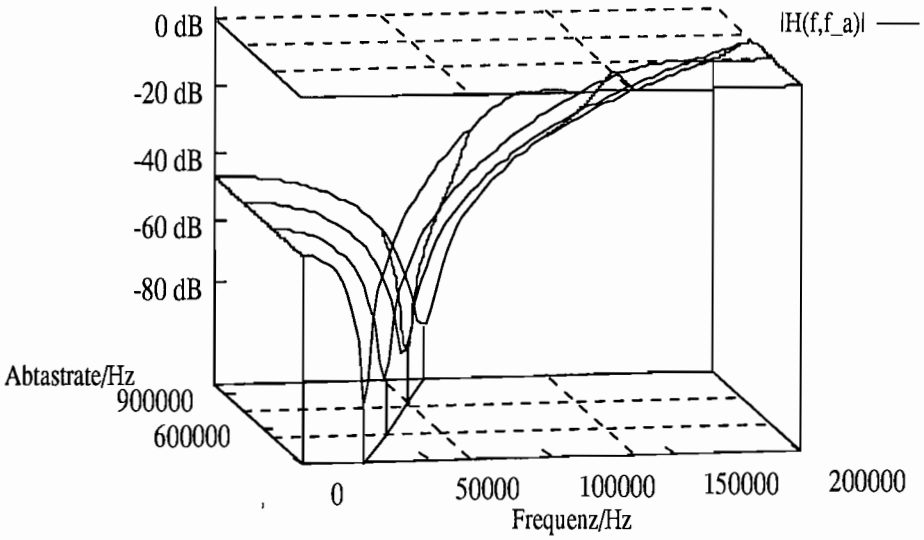


Bild 3.3.1 Durchstimmen der Taktfrequenz bewirkt eine Verschiebung des „Noise Shaping".

Aus den geschilderten Schwierigkeiten ergab sich, daß die Lösung **b)** bevorzugt wurde. Bild 3.3.1 zeigt die Verschiebung des „Noise Shaping" durch alleinige Veränderung der Taktfrequenz. Dieses Filter ist mit Hilfe einer geeigneten Frequenzsynthese, siehe z. B. [MB92b], dann beliebig fein durchstimmbar, lediglich die Selektion der Bandbreite Δf muß durch das digitale Dezimationsfilter in der letzten Stufe erfolgen.

Für das analoge Filter wurden einfache Strukturen nach [UHKI88, S. 1900] und [Can85, S. 251] untersucht. Die Wahl fiel letztendlich auf eine Struktur nach [AS83, S. 1093], die in der klassischen Regelungstechnik unter Frobenius–Form [Unb86, S. 87] bekannt ist, siehe Bild 3.3.2(a), da sie einfach zu realisieren ist und eine bezüglich Stabilität, Eigenschwingungen etc. erprobte regelungstechnische Schaltung darstellt. Weiterhin zeigt sich, daß mit Hilfe dieser Schaltung Pole und Nullstellen weitestgehend unabhängig voneinander einstellbar sind. Um eine beliebige Parameterwahl zu ermöglichen, wurde in [Hut92] ein flexibler Analogteil für einen Wandler 4. Ordnung entworfen, wobei durch Widerstandsverhältnisse die gewünschte Form des „Noise Shaping" einfach eingestellt werden kann.

Für die Berechnung der Filterparameter ist eine Systembeschreibung im z–Bereich nötig. Für die Integratoren setzen wir $H_{\mathrm{I}}(z) = z^{-1}/(1-z^{-1}) = 1/(z-1)$ an.
Die Übertragungsfunktion des Filters im z–Bereich berechnet sich zu

$$H(z) = \frac{V(z)}{W(z)} \tag{3.3.1}$$

Mit

$$V(z) = U(z)\sum_{i=0}^{N} A_i \left(\frac{1}{z-1}\right)^i \tag{3.3.2}$$

und

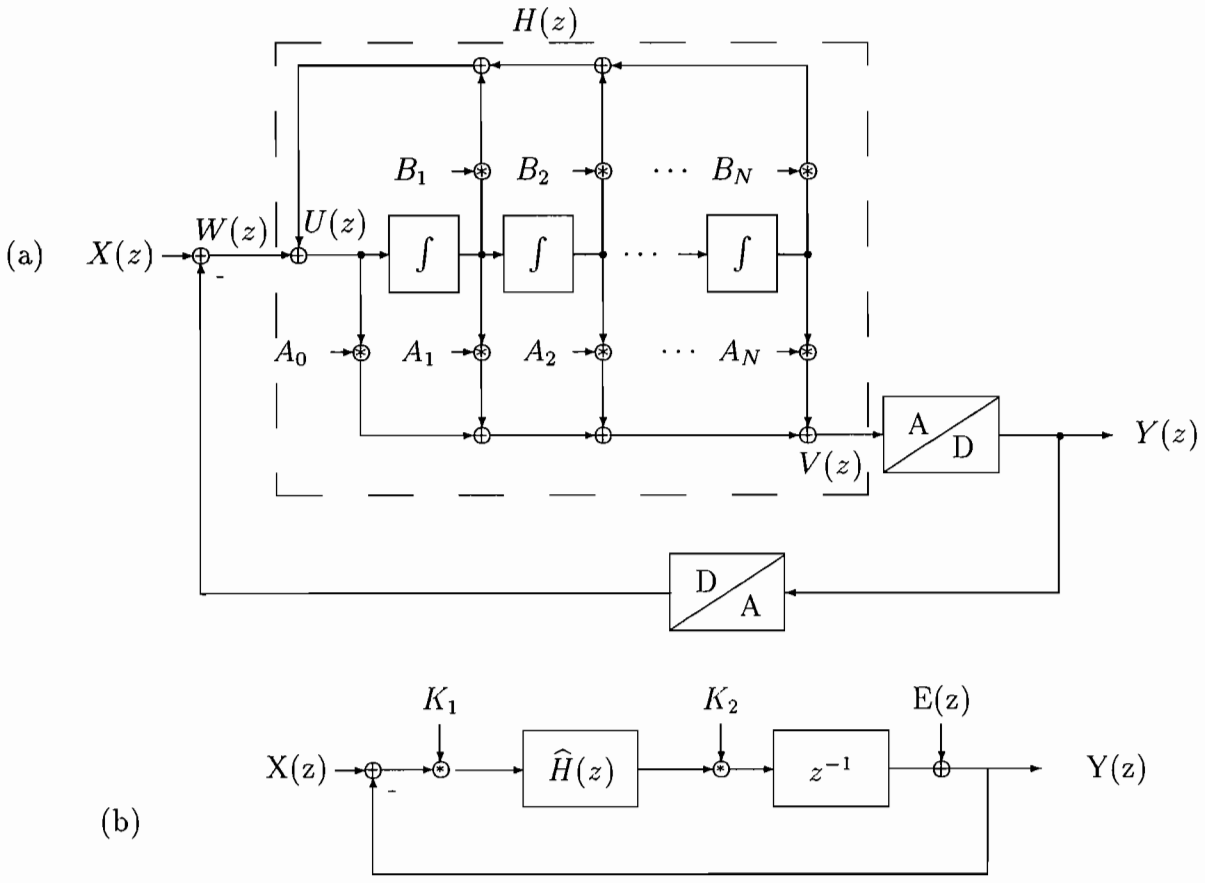


Bild 3.3.2 (a) Filterdarstellung des BP–ΣΔ–A/D–Wandlers in Frobenius–Form. (b) Ersatzschaltbild im z–Bereich.

$$U(z) = \frac{W(z)}{1 - \sum_{i=1}^{N} B_i \left(\frac{1}{z-1}\right)^i} \tag{3.3.3}$$

folgt für die Übertragungsfunktion $H(z)$ des Filters

$$H(z) = \frac{V(z)}{W(z)} = K_1 \widehat{H}(z) \tag{3.3.4}$$

$$= \frac{\sum_{i=0}^{N} A_i \left(\frac{1}{z-1}\right)^i}{1 - \sum_{i=1}^{N} B_i \left(\frac{1}{z-1}\right)^i} \tag{3.3.5}$$

$$= \frac{\sum_{i=0}^{N} A_i (z-1)^{N-i}}{(z-1)^N - \sum_{i=1}^{N} B_i (z-1)^{N-i}} \tag{3.3.6}$$

$$= K_1 \frac{\prod_{i=1}^{N} z - z_{0i}}{\prod_{i=1}^{N} z - z_{\infty i}} \tag{3.3.7}$$

$$= K_1 \frac{Zr(z)}{Nr(z)} \tag{3.3.8}$$

Mit $K = K_1 \cdot K_2$ ergibt sich für die Nutzsignalübertragungsfunktion $H_x(z)$ (STF) bzw. die Störübertragungsfunktion $H_e(z)$ (NTF):

$$H_x(z) = \frac{Kz^{-1}\widehat{H}(z)}{1 + Kz^{-1}\widehat{H}(z)} = \frac{KZr(z)}{zNr(z) + KZr(z)} \tag{3.3.9}$$

$$H_e(z) = 1 - H_x(z) \tag{3.3.10}$$

$$= \frac{1}{1 + Kz^{-1}\widehat{H}(z)} = \frac{zNr(z)}{zNr(z) + KZr(z)} \tag{3.3.11}$$

Die Stabilität des Systems ist gegeben, wenn die Pole der beiden Übertragungsfunktionen $H_x(z)$ und $H_e(z)$ innerhalb des Einheitskreises liegen. Da beide Übertragungsfunktionen den gleichen Nenner besitzen, vereinfacht sich der Entwurf wesentlich.

Beim Bandpaß–Wandler ist allerdings darauf zu achten, daß keine Pole im Nutzfrequenzband liegen, da dies zu einer Verringerung des erreichbaren „Noise Shaping“ führen würde.

Anhand der Gleichungen (3.3.9) und (3.3.11) wird zweierlei deutlich:

1. Die Pole des Filters $H(z)$, d. h. $z_{\infty i}$, werden zu den Nullstellen der Rauschübertragungsfunktion $H_e(z)$.

2. Die Nullstellen des Filters $H(z)$ sind ebenso Nullstellen der Nutzsignalübertragungsfunktion $H_x(z)$.

Dieser 2. Ansatz bietet über die bisherigen Untersuchungen von BP–ΣΔ–A/D-Wandlern hinausgehend eine zusätzliche Freiheit in der Formung des Nutzsignalspektrums. Die bisherigen Untersuchungen beschränken sich auf reine Stabilitätsbetrachtungen der Gesamtübertragungsfunktion [LH93] oder auf eine Maximierung des S/N–Verhältnisses [Dre91]. Zwei Beispiele sollten veranschaulichen, wo dieser zusätzliche Freiheitsgrad nützlich sein kann:

a) Bei LW–Empfängern ist meist ein erhebliches Netzbrummen (bei Vielfachen von 50 Hz oder $16\frac{2}{3}$ Hz) im Antennensignal vorhanden, so daß bis zu Frequenzen von einigen kHz Oberwellen auftreten, siehe Bild 3.3.3(a). Dadurch muß, wenn man eine Übersteuerung vermeiden will, die Eingangsverstärkung klein gehalten werden. Diese Oberwellen können außerdem nach einer Dezimationsstufe in das Nutzsignal gespiegelt werden, wobei das S/N–Verhältnis degradiert würde.

b) Plaziert man die Nullstellen z_{0i} im Bereich der Spiegelfrequenz ($f_a/2$), so vereinfacht sich der Entwurf des Aliasingfilters, da bereits durch den ΣΔ–Analogteil eine Dämpfung in diesem Bereich vorliegt und so die Übergangsbandbreite weiter gewählt werden kann. Der Aufwand des Aliasingfilters wird geringer. Für die Realisierung eines durchstimmbaren BP–ΣΔ–A/D–Wandler ist dies besonders nützlich, da hier das Aliasingfilter erst nach der maximal gewünschten Signalfrequenz beginnen darf und vor der Spiegelfrequenz der minimal gewünschten Signalfrequenz bereits die Sperrdämpfung erreicht sein muß. Bild 3.3.3(b) zeigt diesen Sachverhalt für eine Abtastung mit sechsfacher Signalfrequenz.

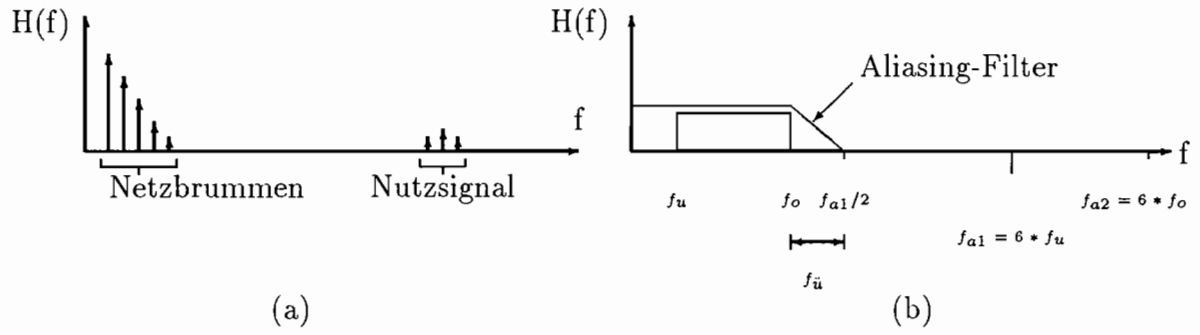


Bild 3.3.3 (a) Typisches Stör- und Nutzsignal eines LW-Antennensignals. (b) Eingangsfilter zur Aliasingunterdrückung.

Im folgenden ist ein Algorithmus für den optimalen Entwurf des BP-ΣΔ-A/D-Wandlers angegeben, der von der Formung des Nutzsignals Gebrauch macht:

Algorithmus : Optimierter BP-ΣΔ-Entwurf

1) Plaziere alle Pole $z_{\infty i}$ von $\widehat{H}(z)$ bei der Signalfrequenz ($\angle z_{\infty i} = 360° f_0/f_a$).

2) Die Menge der variablen Nullstellen M_0 sei leer. Plaziere alle Nullstellen z_{0i} der Menge M_1 von $\widehat{H}(z)$ bei den gewünschten Sperrbereichen der Signalübertragungsfunktion $H_x(z)$ z. B. $\angle z_{0i} = 0°$ oder $\angle z_{0i} = 180°$.

3) Bestimme ein optimales K und alle Nullstellen z_{0i} der Menge M_0 so, daß

- **a)** $|z_{e\infty i}| < 1 \; \forall \, i$ (Stabilität der Gesamtübertragungsfunktion)
- **b)** $S + N < 3$ (Summe aus Nutz- und Störsignal darf Quantisierer nicht übersteuern) †
- **c)** Nenner $H_e(z = e^{j2\pi f_0/f_a}) \rightarrow \mathit{maximal}$ (d. h. Maximierung des S/N-Verhältnisses).

4) Ist **a** oder **b** verletzt, so setze eine weitere feste Nullstelle der Menge M_1 zu der Menge M_0 und weiter mit **3)** sonst **5)**.

5) Nun werden iterativ die $z_{\infty i}$ so optimiert, daß ein Maximum des S/N-Verhältnisses, das proportional zu dem Produkt FP ist, erreicht wird.

Zu den einzelnen Optimierungen sei angemerkt:

Bei der Optimierung nach **3)** kann *kein* Gradientenabstiegsverfahren benutzt werden, da wegen Bedingung **3b)** das Maximum auf dem Rand angenommen wird. Da für den resultierenden S/N-Gewinn eine sehr genaue Berechnung der $z_{e\infty i}$ nicht nötig ist und die konjugiert komplexen Pole symmetrisch zu Nullstellenwinkel auftreten sollten, kann

† Bei Empfängern ist gewöhnlich $S \ll N$, so daß lediglich auf $N < 3$ geprüft werden muß.

eine globale Minimumsuche durchgeführt werden. Ebenso können Heuristiken, wie z. B. „Simulated Annealing", angewendet werden.

Bei Punkt **5)** ist die Plazierung der Nullstellen z_{0i} bereits abgeschlossen. Die Optimierungen zeigten, daß durch diese Verschiebung der $z_{\infty i}$ auf dem Einheitskreis die Bedingung **a)** und **b)** vernachlässigbar beeinflußt werden. Für die Optimierung unter **5)** ist ein Gradientenverfahren anzuwenden, da hier eine genaue Bestimmung für den maximalen S/N–Gewinn nötig ist. Hier wurde die „Downhill"–Methode nach Nelder und Mead [P+92a, S. 408] benutzt.

Im folgenden sind einige Ergebnisse für den vorgeschlagenen Algorithmus für Wandler 2. und 4. Ordnung gezeigt. Die Nutzsignale liegen hierbei bei $f_0/f_a = 1/3, 1/4$ und $1/6$, was einem Nullstellenwinkel von $120°, 90°$ bzw. $60°$ entspricht. Die Nullstellenwinkel wurden so gewählt, da für diese Überabtastverhältnisse besonders leistungsfähige Dezimationsfilter entworfen werden können, siehe Abschnitt 3.4.

Bereits in Abbildung 3.2.6 wurden die erreichten S/N–Werte für verschiedene Überabtastfaktoren gezeigt, wobei hier *keine* Signalsperren berücksichtigt wurden. Für den Wandler 4. Ordnung konnte durch alleinige Optimierung der Nullstellenlage (bzw. Pole von $\widehat{H}(z)$) rund 3 dB für alle Überabtastfaktoren gewonnen werden. Abbildung 3.2.5 zeigt die zugehörigen Signalübertragungsfunktion $H_x(f)$ und Rauschübertragungsfunktion $He(f)$ für verschiedene Nutzsignal–Bandbreite. Abbildung 3.2.6 zeigt auch, daß beim Wandler 2. Ordnung nur eine freie Nullstelle im Signalfrequenzbereich liegt, so daß die Plazierung unter **1)** bereits optimal ist. Nun sollen noch einige konkrete Pol/Nullstellenlagen gezeigt werden.

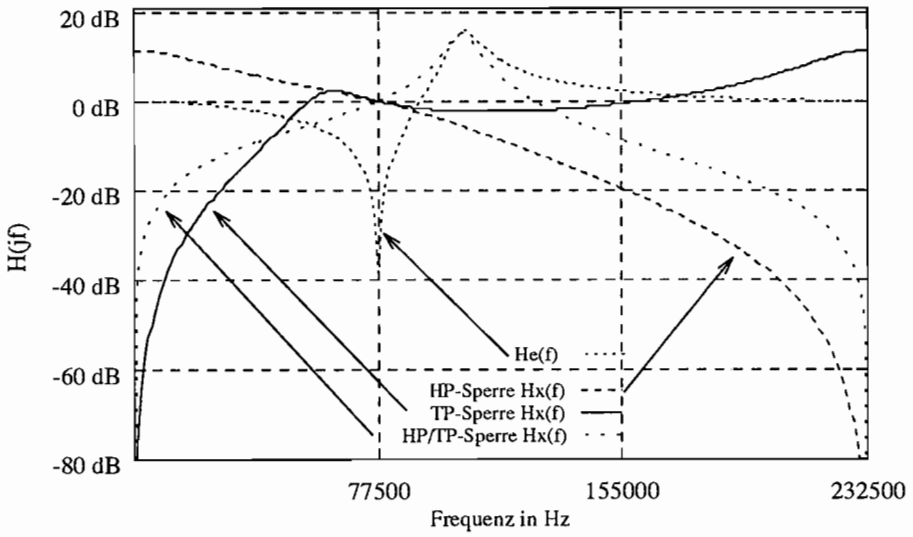


Bild 3.3.4 Wandler 2. Ordnung mit TP,HP und TP/HP–Bandsperren.

Zunächst zeigen die Bilder 3.3.5(a)–(d) die Ortskurven, d. h. mögliche Lagen von Polen und Nullstellen, bei einer gewünschten TP–, HP–, TP/HP–Signalsperre und ohne Signalsperre für das Nutzsignal. Die Ortskurven zeigen den stabilen Bereich der Gesamtübertragungsfunktion, wobei zusätzlich auch die Bedingung $N < 3$ eingehalten ist. Für die zugehörigen Stabilitätsbereiche kann mit Hilfe des Schur–Cohn–Jury Kriterium der Wertebereich der Verstärkung K bestimmt werden. Es ergibt sich $0 \leq K \leq 0,75$ für den Nullstellenwinkel $60°$, $0 \leq K \leq 0,5$ für den Nullstellenwinkel $90°$ und $0 \leq K \leq 0,25$

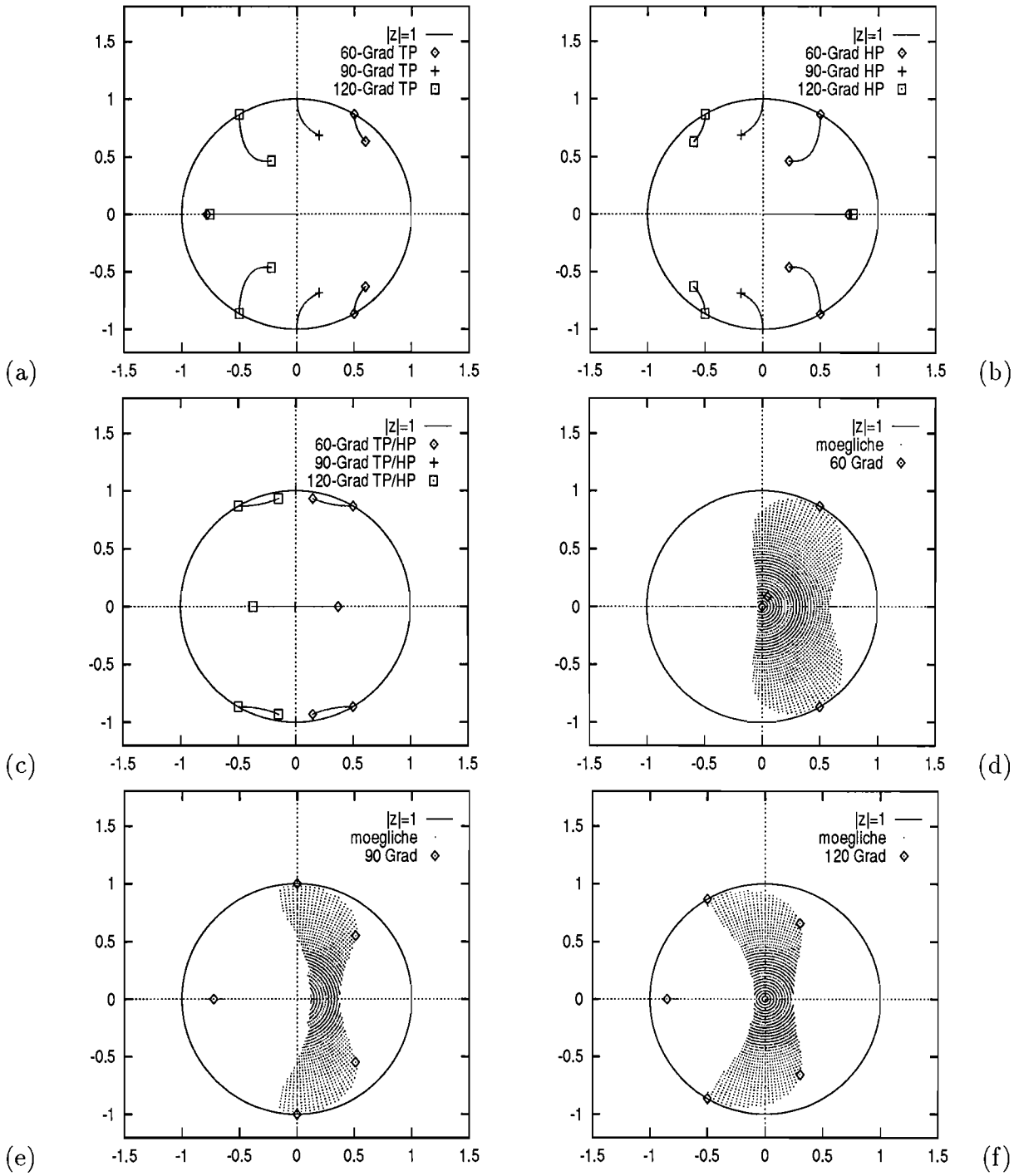


Bild 3.3.5 Mögliche (·) und optimale (◇) Pol/Nullstellenverteilung für 60, 90 und 120 Grad FSF–Filter. Erläuterung siehe Text.

für den Nullstellenwinkel 120° bei der TP/HP–Aufteilung. Dies stimmt mit der durch Simulation bestimmten Größe für die Schleifenverstärkung nach [Dre91] überein, obwohl dort die durch den Abtaster vorhandene Verzögerung z^{-1} *nicht* berücksichtigt wurde. In Bild 3.3.4 sind für einen Wandler 2. Ordnung die für TP–, HP– und HP/TP–Signalsperren entworfenen Stör– und Nutzsignalübertragungsfunktionen darge-

stellt. Deutlich ist die Dämpfung bei $f = 0$ für TP– und TP/HP–Sperre und die Dämpfung bei $f = f_a/2 = 3 \cdot 77{,}5$ kHz für HP– und TP/HP–Optimierung zu erkennen.

Interessant ist weiterhin, daß sich das S/N–Verhältnis für Nullstellenwinkel von 60° und 120° durch die Signalsperren nur unwesentlich (< 1 dB) gegenüber einer beliebigen Plazierung der z_{0i} verschlechtern wird. Bei dem Nullstellenwinkel von 90° und TP/HP–Optimierung ($Zr(z) = (z-1)(z+1)$) allerdings müssen die Pole, um Stabilität und $N < 3$ einzuhalten, sehr dicht an die Nullstellen rücken, so daß hier ein erheblicher S/N–Verlust auftritt, siehe Bild 3.3.5(c).

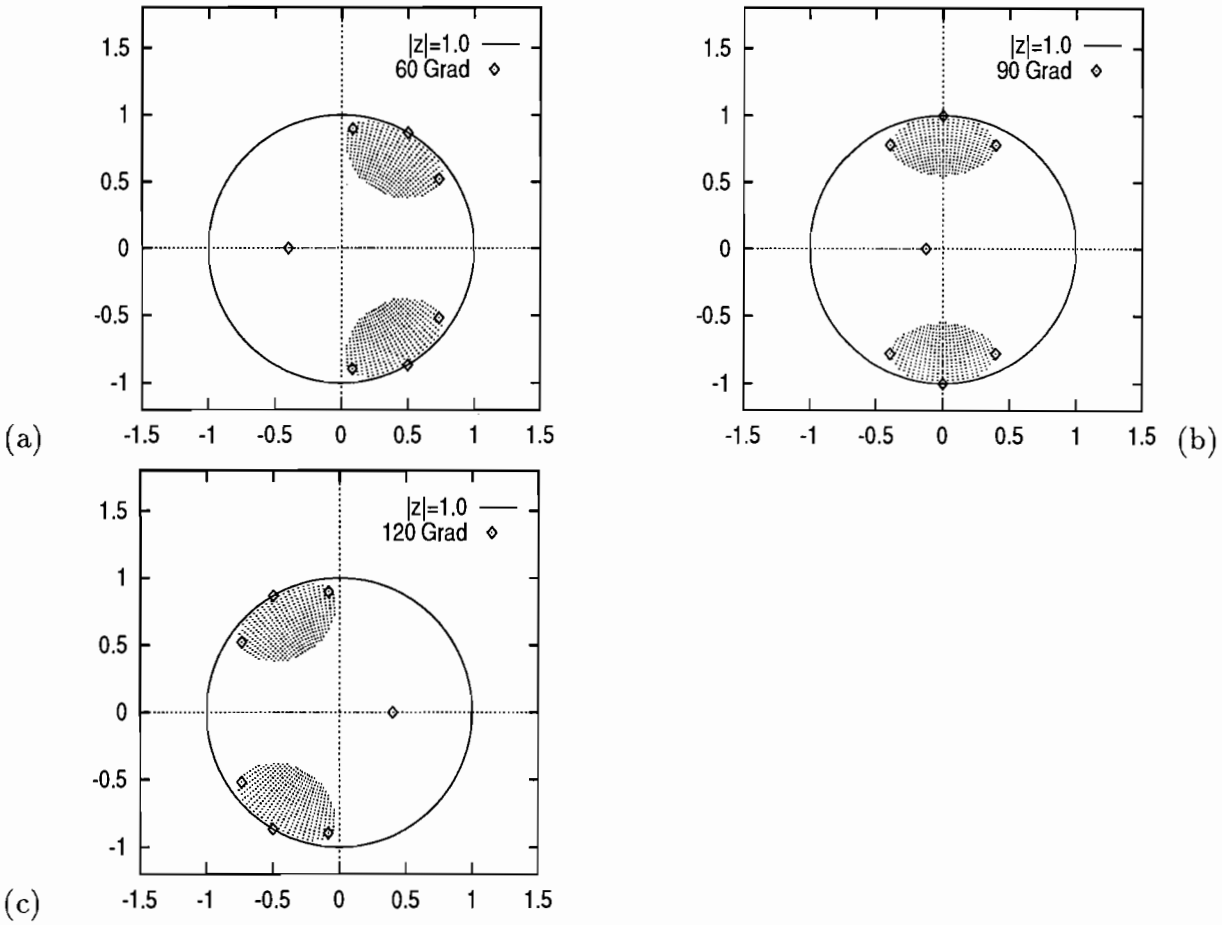


Bild 3.3.6 Mögliche(·) und optimale (◇) Pol/Nullstellenverteilung für 60, 90 und 120 Grad FSF–Filter für Wandler 4. Ordnung ohne Bandsperren.

In den Bildern 3.3.6(a)–(c) sind die möglichen und optimalen stabilen Bereiche für den Wandler 4. Ordnung ohne Bandsperren gezeigt. Es ist wiederum zu erkennen, daß das Maximum auf dem Rand des erlaubten Bereichs angenommen wird.

3.3.1 DCF77–BP–ΣΔ–A/D–Wandler

Als konkretes Beispiel sollen nun die Ergebnisse des Entwurfs BP–ΣΔ–Analogteil für DCF77–Signale gezeigt werden [Hut92].

Bei einer Signalfrequenz von 77,5 kHz und einer Abtastfrequenz von 12·77,5 kHz ≈1 MHz soll eine Bandbreite von ca. 5 kHz (Abtastrate nach der letzten Dezimation

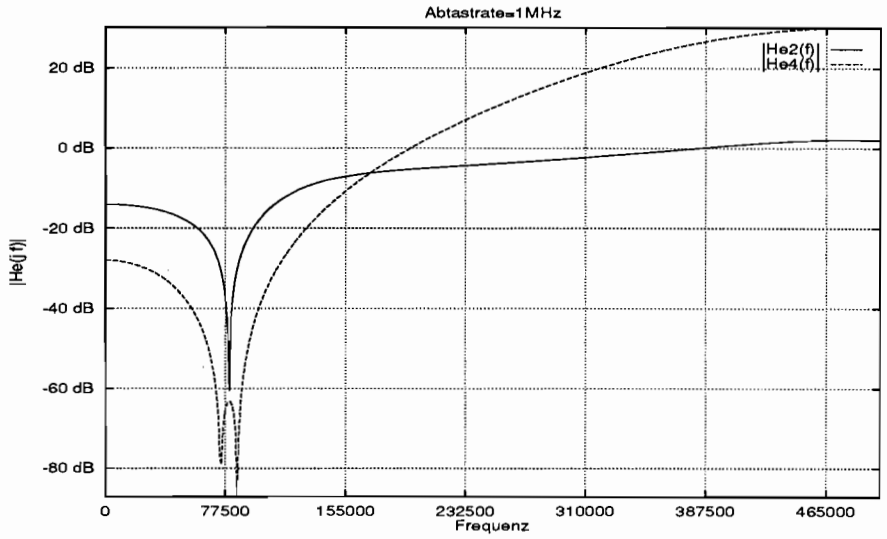


Bild 3.3.7 Rauschübertragungsfunktion eines BP-ΣΔ-A/D-Wandler 2. und 4. Ordnung für DCF77-Signale.

größer 10 kHz) realisiert werden. Aus dem Diagramm von Bild 3.2.6 ergibt sich bei der Oversampling-Rate von 1 MHz/10 kHz=100 eine zu erwartende Signal/Rauschleistung am Ausgang der Dezimation von 60 dB für einen Wandler zweiter Ordnung und ca. 85 dB, d. h. 14 Bit, für einen Wandler vierter Ordnung ab. In Bild 3.3.7 sind die Rauschübertragungsfunktionen für die BP-ΣΔ-A/D-Wandler 2. bzw. 4. Ordnung gezeigt. Das Bild 3.3.8 zeigt das mit einer 4096-Punkte-FFT und vier Mittelungen gemessene Spektrum des Wandlers vierter Ordnung für ein sinusförmiges Eingangssignal der Frequenz 77,5 kHz [Hut92].

Das Bild 3.3.8 verdeutlicht eindrucksvoll, wie das Rauschspektrum durch „Noise Shaping" geformt wird. Die zu erwartende hohe Auflösung des A/D-Wandlers wird trotz des

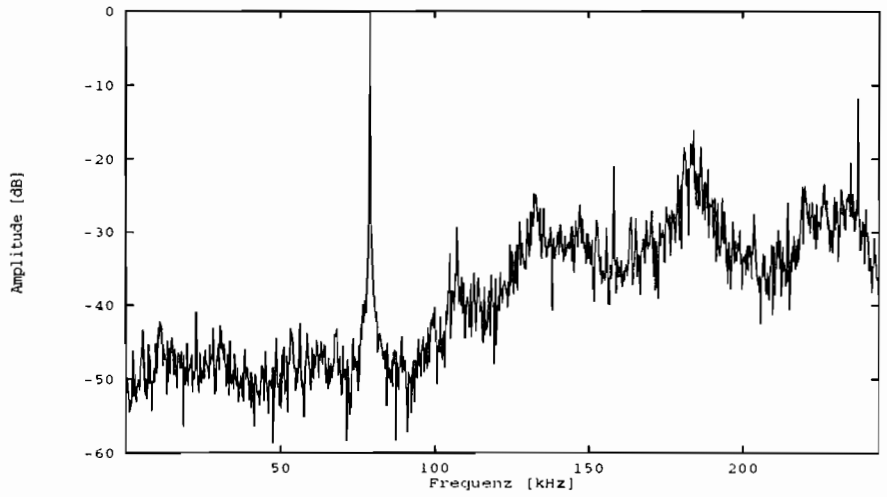


Bild 3.3.8 Gemessenes Spektrum für den BP-ΣΔ-A/D-Wandler 4. Ordnung für DCF77-Signale.

schmalbandigen Eingangssignals in Bild 3.3.7 ohne Dezimationsfilter nicht vollständig deutlich. Ziel ist es daher, im nächsten Abschnitt ein dezimierendes Filter mit möglichst geringem Aufwand zu entwerfen.

3.4 Dezimationsfilter für den ΣΔ–Bandpaßwandler

Der Entwurf eines digitalen Dezimationsfilters dient zum einen dazu, das Summenspektrum aus Nutzsignal und das durch „Noise Shaping" geformte Rauschspektrum so zu filtern, daß ein hohes S/N–Verhältnis entsteht, zum anderen dazu, eine nachfolgende Auswertung des A/D–gewandelten Signals, z. B. Demodulation durch Signalprozessor, durch eine Reduktion in der Abtastrate stark zu entlasten. Es wird sich zeigen, daß diese Filter besonders günstig mit einem programmierbaren Gate Array (PGA) realisiert werden können, da so eine Realisierung in einer Pipeline–Struktur möglich ist und außerdem flexibel die Bit–Breiten einer kaskadierten Dezimationsstufe den geforderten S/N–Verhältnissen angepaßt werden können. Selbst der leistungsfähigste DSP könnte diese Filter nicht realisieren, da er „nur" *einen* schnellen Multiplizierer besitzt.

Liegt eine Abtastung oberhalb der Nyquistfrequenz vor, so empfiehlt es sich, in den ersten Stufen Halbbandfilter zur Dezimation um einen Faktor 2 einzusetzen, da diese mit sehr geringem Aufwand, d. h. geringem Platzbedarf im PGA, realisiert werden können, siehe z. B. [CR83, S. 231-5] und [Sch91b].

Für die Realisierung der hohen Dezimation ($M \gg 2$) des Digitalfilters erscheinen drei Lösungsmöglichkeiten geeignet:

1) (Komplexer) Quadraturmischer, gefolgt von zwei TP–Filtern

2) Hilbert–Transformator, gefolgt von zwei TP–Filtern

3) Bandpaßdezimator

Die Quadraturmischer–Lösung ist in der Literatur ausführlich behandelt worden ([Wie88], [Dre91], [SJF92]) und die Hilbert–Transformator–Lösung wird ausführlich im Abschnitt 4.1.4 behandelt. Die Realisierungen **1)** und **2)** beschränken sich im wesentlichen auf den Entwurf eines leistungsfähigen TP–Filters. Für die Realisierung von leistungsfähigen und aufwandsgünstigen Dezimationsfiltern für den TP–Bereich sind die Halbbandfilter [CR83] für einen Wechsel der Abtastrate um den Faktor zwei und CIC–Filter ([Hog81], [AW91]) wohl bekannt. Für die Dezimation geeignete schmalbandige BP–Filter entsprechend Lösung **3)** sind Filter, die einen mit den CIC–Filtern vergleichbaren, geringen Hardware–Aufwand besitzen, und sind bisher aus der Literatur nicht bekannt [OS92], [SH90], [CR83]. In der Folge wird eine neue Klasse von schmalbandigen BP–Filtern vorgestellt und analysiert, die eine ideale Realisierung eines Bandpaßdezimators für ΣΔ–Bandpaßwandler erlauben.

Es ist aus der Literatur bekannt ([SH90, S. 295], [KK89, S. 68]), daß Frequenzabtastfilter (FSF) besonders aufwandsgünstige Realisierungen von Dezimations–/Interpolationsfiltern für Bandpaßsignale darstellen. Weiterer Komplexitätsgewinn kann durch die Einführung von Multiplikationen mit CSD–Koeffizienten† ([GC77], [PF88], [HAHL85])

† Das pseudoternäre Zahlensystem stellt eine Erweiterung des binären ($b_i \in [\pm 1]$) Systems einer B Bit Zahl $X = \sum_{i=0}^{B-1} b_i 2^i$ auf das dreiwertige System $X = \sum_{i=0}^{B-1} t_i 2^i$ mit $t_i \in [-1, +1, 0] = [\bar{1}, 1, 0]$

oder durch Umstrukturierung des Filters (vorgezogene Reduktion), die bei TP-Signalen als CIC-Filter [Hog81] bekannt ist, erreicht werden, siehe Bild 3.4.1. Es soll nun untersucht werden, welche rekursiven Elementarfilterstrukturen (BiQuad) für die Umstrukturierung bei BP-Filtern geeignet sind. Als günstige Strukturen bei endlichen Registerbreiten und Überlaufarithmetik in Bezug auf Stabilität sind „State Space Filter" (Minimum Norm Filter [BF77]), Wellendigitalfilter (WD [MEA89], [ML78], [Fet86], [MM80]) und verschiedene Strukturen mit geringer Empfindlichkeit bezüglich Koeffizientenquantisierung ([DA85], [NHP81]) bekannt. Da FSF zur Klasse der FIR-Filter gehören (obwohl sie rückgekoppelte Filter sind!), wird für die Stabilität des Filters — außer der BIBO-Stabilität und Grenzzyklenfreiheit — die Äquivalenz zur nicht rekursiven Realisierung gefordert. Die Umstrukturierung der FSF-Filter wie bei CIC-Filtern ist bezüglich der Stabilität des Filters nicht unkritisch, da

- die Stabilität bei Filtern mit nichtlinearen Überläufen nicht invariant gegen Umstrukturierungen ist [BF77, S. 573]
- keine allgemeine Stabilitätsaussage für beliebige Filter bei „nonzero input" möglich ist [Bro90, S. 891]

Aus der Vielzahl der aus der Literatur bekannten, nicht stochastischen Stabilitätsuntersuchungen ([CMP76a, Tab I], [EMT69]) befaßten sich die meisten mit der „Zero Input Overflow Limit Cycle" (ZIOLC) Analyse. Hier sind oftmals „State Space Filter" ([Bro90], [ZL81], [MMR78], [Sin86]) untersucht worden. Bei den Nichtlinearitäten wurden Lyapunov-Analysen ([CMP75], [SP83], [Ur81], [Bol84]) durchgeführt, die nur beim Finden einer geeigneten Lyapunov-Funktion zum Erfolg führen. Außerdem wurde das „Jump Phenomen" ([Kri74], [CK74]) analysiert. Für ein in der Praxis einzusetzendes Filter ist sicherlich die ZIOLC-Freiheit notwendig, zu fordern ist aber in *jedem Fall* eine „Force Response Stability" (FRS). Dieser Stabilitätstest ist sehr aufwendig und nur in einigen Spezialfällen deterministisch lösbar [CMP76b], [Aue88], [Bro90].

In den folgenden Abschnitten wird untersucht, wie sich kanonische, realisierbare Elementarfilter (d. h. minimale Anzahl an Registern bzw. keine WD) bezüglich ZIOLC und FRS bei einer Kaskadierung verhalten. Zwei neuartige Elementarfilter werden beschrieben, und der detaillierte Entwurf der einzelnen Registerbreiten in der kaskadierten Form wird besprochen.

3.4.1 Stabilitätstests für Frequenzabtastfilter

Wie aus der Literatur bekannt, können zwei Fehlerquellen unterschieden werden:

1) Quantisierungsfehler durch Koeffizienten endlicher Genauigkeit

2) Nichtlineare Operationen durch Arithmetiküberläufe

Fehler vom Typ **1)** können durch ausreichend große Bit-Breiten weitestgehend vermieden werden, während Arithmetiküberläufe nur durch spezielle Filterstrukturen egalisiert werden können.

---

und $t_i \cdot t_{i+1} = 0$ dar, siehe z B. Bild 3.4.9. In der englischsprachigen Literatur ist die Bezeichnung CSD-Code (canonically-signed-digit code) üblich [Hes89, S. 162-3].

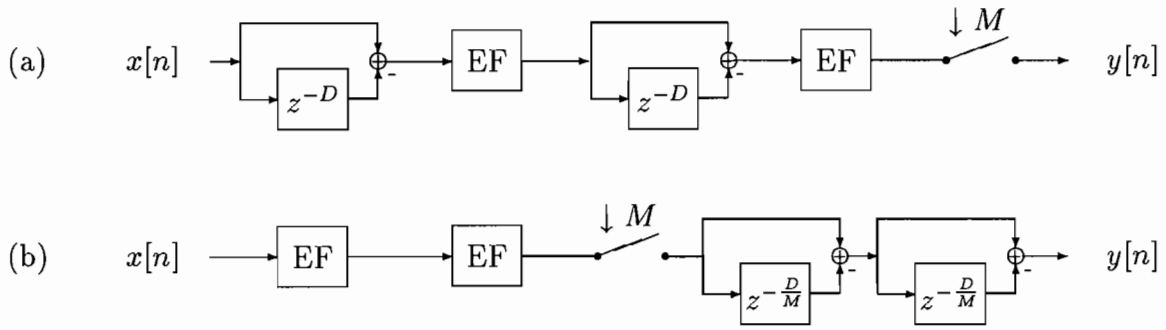


Bild 3.4.1 (a) Dezimator als Kaskade aus 2 Frequenzabtastfiltern (b) Umgestelltes CBPC–Filter zur Dezimation.

Betrachten werden sollen kaskadierte FSF–Elementarfilter zur Dezimation, siehe Bild 3.4.1(a). Eine weitere Aufwandsreduktion soll durch eine Umgruppierung entsprechend Bild 3.4.1(b) erreichen werden, wodurch die Anzahl an benötigten Verzögerungsgliedern um einen Faktor R verringert wird. Diese Struktur soll in Analogie zum CIC–Filter mit CBPC (cascaded bandpass comb) bezeichnt werden.

Aus der Systemfunktion $H(z)$ für das FSF–Filter

$$H(z) = \frac{1}{N}\left(1 - z^{-D}\right) \sum_{K<N/2} (-1^K) H_K \frac{1 - z^{-1}\cos(2\pi K/D)}{1 - 2z^{-1}\cos(2\pi K/D) + z^{-2}} \qquad (3.4.1)$$

entnehmen wir, daß an den Stellen $H_k \neq 0$ eine Pol/Nullstellenkompensation auftritt. Diese Pol/Nullstellenkompensation ist bezüglich Fehlern der Arten **1)** und **2)** sensitiv.

Da FSF–Filter zur seltenen Klasse der *rückgekoppelten* Filter mit *endlicher Impulsantwort* (FIR–Filter) gehören, vereinfacht sich der Stabilitätstest wesentlich. Wie eingangs erwähnt, wird die funktionale Äquivalenz zu der nichtrückgekoppelten FIR–Struktur gefordert. Für ein korrektes Funktionieren des Filters ist es dabei nötig, daß die „wrap arounds“ in der K2–Arithmetik, ebenso wie bei CIC–Filtern, durch die nachfolgenden Differenzierer wieder ausgeglichen/egalisiert werden. Da eine funktionale Äquivalenz zum nichtrekursiven FIR–Filter angestrebt ist, kann keine Sättigungslogik o. ä. (siehe z. B. [EMT69, S. 3017] oder [CMP76b, S. 695]) zur Verbesserung der Stabilitätseigenschaften eingesetzt werden. Folgende 3 Stabilitätstest erscheinen für CBPC–Filter sinnvoll:

a) „Forced Response Stability“ bei Eigenfrequenzanregung

b) „Zero Input Limit Cycle“–Test

c) „Forced Response Stability“ bei beliebiger Eingangsfolge

zu a) Die stärkste Aussteuerung für ein BP–Filter ist dann gegeben, wenn es mit seiner Eigenfrequenz angeregt wird. Hier wird besonders schnell Fehlverhalten im Überlauf deutlich. Hierzu muß insgesamt nur ein Test durchgeführt werden.

zu b) Gehen wird davon aus, daß das Filter einen beliebigen Register–Inhalt annehmen kann §, so ist für einen vollständigen ZIOLC (der im allgemeinen bei IIR–Filtern gar nicht durchgeführt werden kann!) ausreichend, daß das Filter bei einem aus Nullen bestehend Eingangssignal nach der Dauer der Impulsantwort ebenfalls eine Nullfolge am Ausgang erzeugt. Bei $B_{\max}$ Bit je Elementarfilter ergeben sich für eine kanonische Struktur mindestens $2^{2NB_{\max}}$ Tests¶, da die Differenzierer rein nichtrekursiv sind und damit in jedem Fall FRS stabil sind und somit *nur* die Registerzustände der rückgekoppelten Struktur, d. h. die N Elementarfilter getestet werden müssen.

zu c) Der vollständige Test bei beliebiger Eingangsfolge bedeutet, daß, ausgehend von einem willkürlichen Startzustand und einer beliebigen Eingangsfolge, die natürlich im erlaubten Aussteuerungsbereich bleiben muß, die Differenz $h_{\mathrm{CBPC}}[k]-h_{\mathrm{FIR}}[k] = e[k] = 0$ $\forall k > N \cdot D$ Null wird, d. h. nach der Dauer der Impulsantwort sich als Differenzfolge eine Nullfolge ergibt. Zur Veranschaulichung des vollständigen FRS–Test diene Bild 3.4.2.

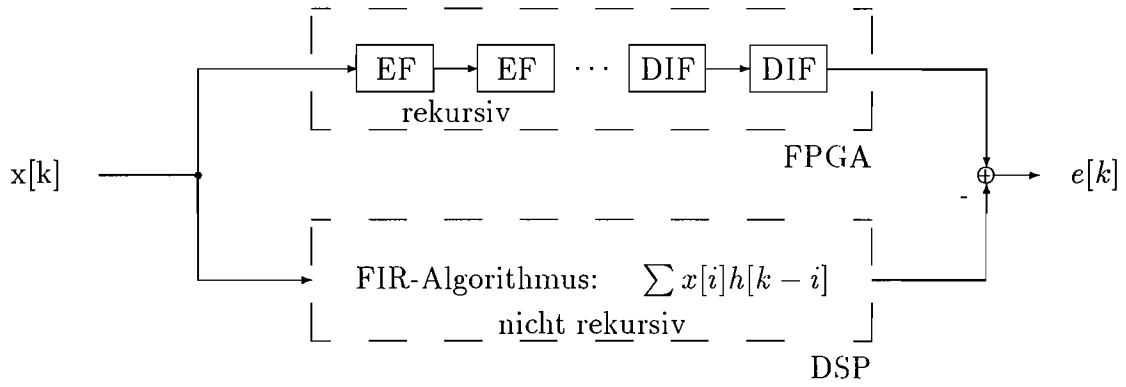


Bild 3.4.2 Realisierung des vollständigen FRS–Test mittels einer parallelen DSP/FPGA–Struktur [RS93].

Für die kanonischen Strukturen ergibt sich hierbei eine Anzahl von mindestens $W^{N\cdot D}2^{2NB_{\max}}$–Tests, wobei W den Wertevorrat der Eingangsfolge $x[k]$ beschreibt. Dieser sehr rechenintensive Test ließ sich, wie in Abschnitt 3.4.2 gezeigt ist, nur durch eine parallele Hardware aus FPGA (rekursiver Teil) und DSP (nichtrekursiver Teil) realisieren.

3.4.2 Stabilitätsuntersuchungen verschiedener Elementarfilterstrukturen

Die direkte Umsetzung von Gleichung (3.4.1) in eine Filterstruktur ist in Bild 3.4.3 dargestellt. Die Cosinus–Faktoren lassen sich noch entsprechend Bild 3.4.3(b) zusammenfassen,

§ Die angenommenen Register–Zustände können mittels eine Hardware–Verifikation, wie z. B. „LOVERT“ [Eve93] bestimmt werden.

¶ Die genaue Anzahl hängt von der in der Impulsantwort zusätzlich vorhandenen Verzögerungen ab.

so daß letztlich eine Schiebeoperation (Multiplikation mit 2) und eine Multiplikation nötig sind [Sch91b, S. 56]. Diese **EFilter**–Struktur besitzt den Vorteil, daß die Verstärkungsfaktoren H_K der verschiedenen Durchlaßfrequenzen gleich sind. Für die Realisierung

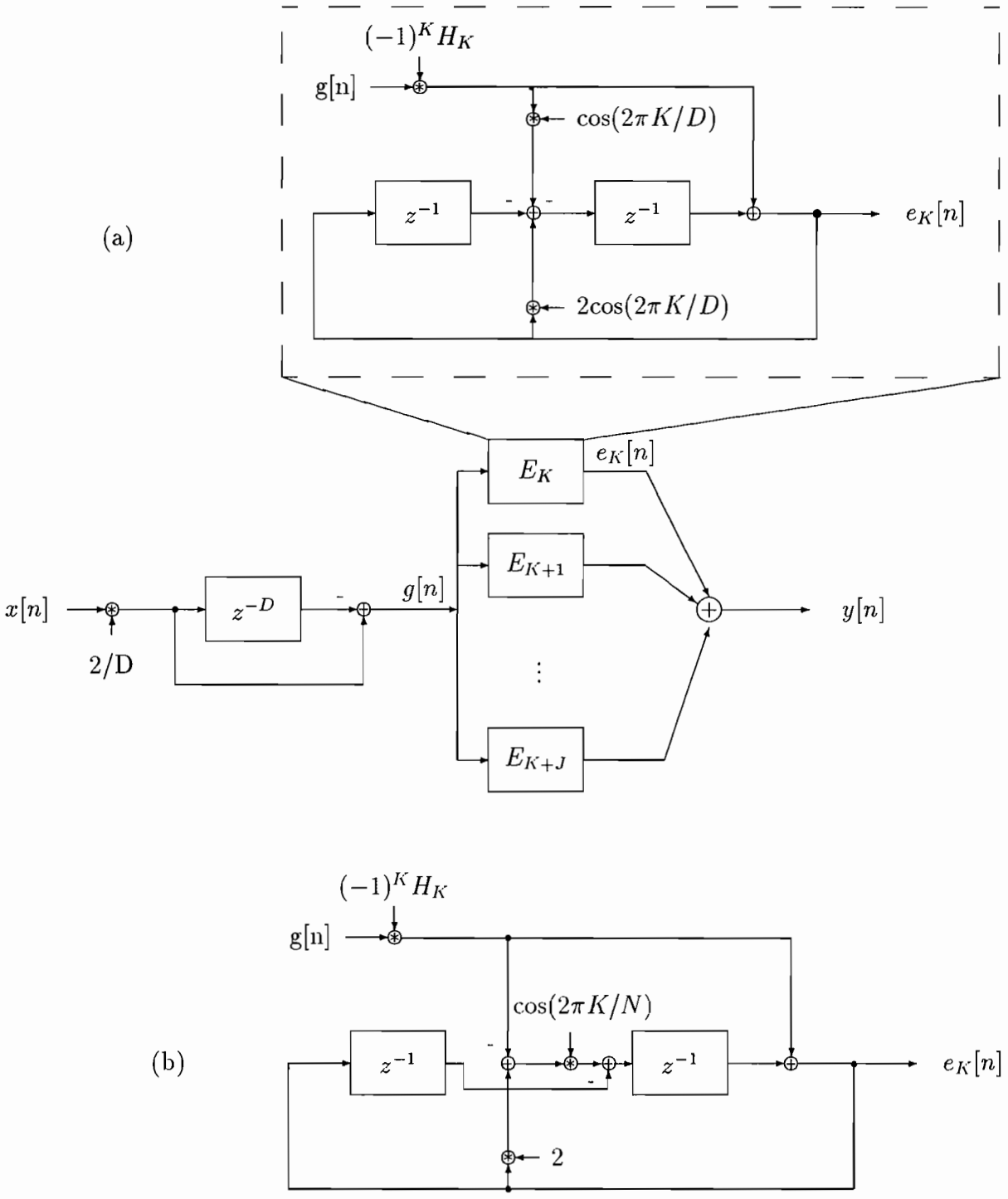


Bild 3.4.3 (a) Frequenzabtastfilter. (b) Modifiziertes Elementarfilter (EFilter).

mit nur einem Koeffizienten, die im folgenden betrachtet wird, kann die Multiplikation mit $(-1)^K H_K$ entfallen. Um zwischen Fehlern der 1. und 2. Art (siehe Abschnitt 3.4.1) unterscheiden zu können, wollen wird eine zweistufige CBPC-Struktur mit EFilter bei $K/D = 1/6 \rightarrow 2\cos(2\pi K/D)=1$ und Eigenfrequenzanregung betrachten, siehe BildBl3.2c.

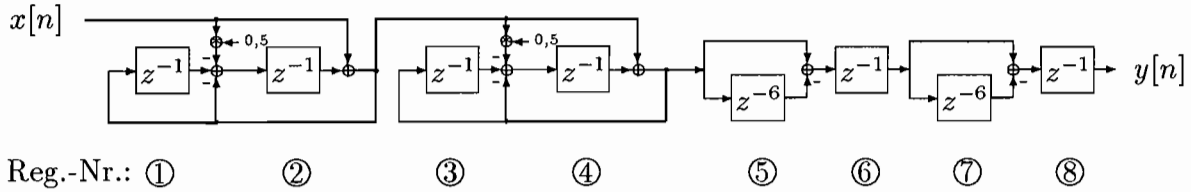


Bild 3.4.4 EFilter-Struktur mit Fehlerverhalten entsprechend Bild 3.4.5.

In Bild 3.4.5(a) ist das Fehlverhalten dieses Filters deutlich zu erkennen. Das Fehlverhalten beruht, da der Koeffizient exakt realisierbar ist, auf einem Fehler im Überlaufverhalten. In Bild 3.4.5(b) sind Über- und Unterläufe der einzelnen Register dargestellt. Die Untersuchung der Sprungantwort (Eigenfrequenztest) bei CIC-TP-Filtern zeigt, daß die Überläufe der Register im rekursiven Teil (in die gleiche Richtung) nur *einmal* im Bereich der Differenziererverzögerung (hier $D = 6$) auftreten dürfen. Anhand des Bildes 3.4.5(b) wird deutlich, daß das Fehlverhalten** ursächlich mit dem unmittelbar nacheinander stattfindenden Überlauf im Register „2" zusammenhängt. Die zweimalige Einkopplung des Eingangssignals bewirkt einen gleichzeitigen Überlauf in Register „2" und den Unterlauf in dem Register „1". Danach läuft Register „2" einen Takt später nochmals über. Der nächste Überlauf (in die gleiche Richtung) hätte aber frühestens nach $D = 6$ Takten auftreten dürfen. Wählen wir hingegen nur eine Einkopplung für das Elementarfilter, wie in Bild 3.4.6(a)(b) oder (c), so zeigt sich kein Fehlverhalten bei Eigenfrequenztest. Im

** Eine Möglichkeit, das Filter zu stabilisieren, wäre die Einführung eines Dämpfungsfaktors $r < 1,0$, d. h. $H(z) = 1/(z^2 - rz\cos(2\pi K/D) + r^2)$. Wie in [BF77, S. 572] gezeigt, ist aber bereits für ZIOLC-Stabilität eine starke Einschränkung der möglichen Pollagen auf ein linsenförmiges Gebiet in der z-Ebene gegeben, was das Dämpfungsverhalten des Filters ganz wesentlich verschlechtern würde und deshalb nicht weiter untersucht werden soll.

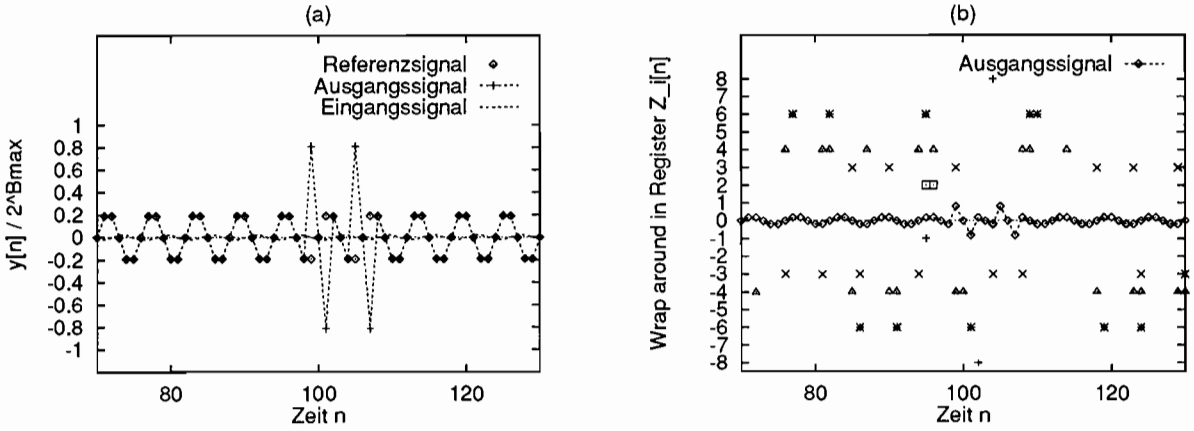


Bild 3.4.5 (a) Fehlverhalten beim EFilter bei Eigenfrequenzanregung (b) Zugehörige Register Überläufe (Y-Achse: $1, 2 \ldots$) und Unterläufe (Y-Achse: $-1, -2 \ldots$). Zu Nummerierung der Register siehe 3.4.4.

Gegensatz zum ZIOLC–Test bei „State Variable Filter“ mit $x_{n+1} = Ax_n + Bg_n$ (siehe z. B. [BF77]), bei dem die Stabilität nur von der Rückkopplungsmatrix A abhängt, ist die Stabilität beim FRS–Eigenfrequenztest hier von der Einkopplung B abhängig.

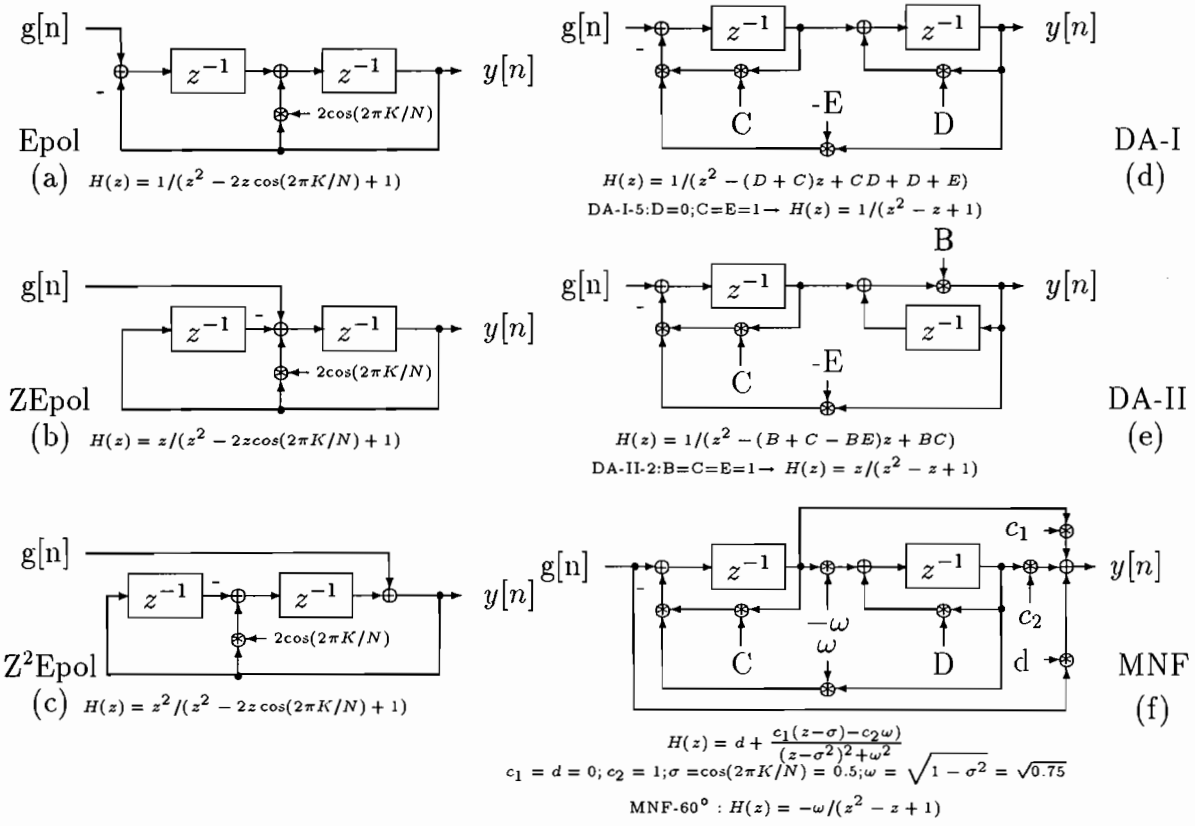


Bild 3.4.6 Die Elementarfilter: Epol, ZEpol, Z²Epol, DA-I-5 DA-II-2 und MNF.

Aus der Literatur sind zwei weitere Strukturen bekannt, die ebenfalls den Eigenfrequenztest für $K/D = 1/6$ bestehen. Dies sind die in Abbildung 3.4.6(d) und (e) dargestellten Strukturen DA-I-5 und DA-II-2, siehe Tabelle 3.4.1.

Das Minimum Norm Filter (MNF), siehe 3.4.6(f) ist zwar für alle Pol/Nullstellenverteilungen ZIOLC–stabil, aber als Elementarfilter im CBPC–Strukturen bei FRS–Eigenfrequenztest versagt es. Da nur eine Einkopplung für das MNF–Filter vorgesehen wurde, ist aber hier das Fehlverhalten auf Fehler vom Typ 1, d. h. Koeffizientengenauigkeit, zurückzuführen. Da der multiplikative Faktor $\omega = 0,866$ nicht mit unendlicher Genauigkeit realisiert werden kann, kommt es zu einer unvollständigen Auslöschung von Pol und Nullstelle. Da andererseits die Polempfindlichkeit von der vektoriellen Differenz der Pole in der z–Ebene abhängt [Hes89, S. 133], resultiert für die kaskadierte Struktur aus Elementarfiltern eine unendliche Polempfindlichkeit. Dies erklärt, warum das MNF nicht stabil arbeitet, und zum anderen, warum nur exakt realisierbare cos–Faktoren der Strukturen aus dem Bild 3.4.6 den Eigenfrequenztest bestehen. Als Polwinkel können somit nur 0° (CIC,CTPC); 60°, 90°, 120° CBPC und 180° CHPC in den fünf stabilen Realisie-

rungen Epol, ZEpol, Z^2Epol, DA-I und DA-II realisiert werden. Die Tabelle 3.4.1 zeigt alle durchgeführten Stabilitätstests.

Struktur	Stufen	K/D	ZIOLC–Test	Sinus–FRS	bel. FRS
EF	1	1/6	√	√	
	2-4	1/6	*Fehler*	*Fehler*	
MNF	1	1/4	√	√	
	2-4	1/6	*Fehler*	*Fehler*	
DA-I-5	1,2	1/6,2/6	√	√	
	3,4	1/6,2/6		√	
DA-II-2	1,2	1/6,2/6	√	√	
	3,4	1/6,2/6		√	
Epol	1,2	1/6,2/6	√	√	
	3,4	1/6,2/6		√	
ZEpol	1,2	1/6,2/6	√	√	
	4	0/12,2/12,3/12,4/12,6/12		√	
	4	1/6,6/36,10/60,600/3600		√	
	2	1/6	√	√	√

Tabelle 3.4.1 Die durchgeführten Stabilitätstest der CBPC–Struktur.

Der aufwendigste Test ist der „Forced Response Stability"–Test bei beliebiger Eingangssequenz. Es werden nur $\Sigma\Delta$–förmige Eingangssignale ($+1, -1 \rightarrow$ Wertevorrat $W = 2$), wie sie am Eingang des BP–$\Sigma\Delta$–A/D-Wandler auftreten, untersucht. Gewählt wurde eine zweistufige ZEpol–Realisierung, da der ZEpol eine relativ kurze Impulsantwort besitzt, aber nur unwesentlich mehr Addiererlaufzeiten benötigt als ein Epol–Filter, siehe Tabelle 3.4.2. Aus der Tabelle 3.4.3 ergibt sich für $K/D = 1/6$ ein Bit–Wachstum $\mathrm{ld}(\sum_i |h_i|)$ von 4, so daß $2^{4 \cdot 5}$–FF–Anfangszustände berücksichtigt werden müssen. Mit einer Impulsantwortlänge von 11 müssen somit $2^{12} \cdot 2^{20} = 2^{32}$ Tests durchgeführt werden. Eine SUN–Sparc–3 realisiert ca. 10^6 Tests pro Tag, so daß die SUN für die 2^{32} Tests rund 11,7 Jahre benötigt. Mit der parallelen FPGA/DSP–Lösung aus Bild 3.4.2 werden ca. 800000 Simulationen pro Sekunde erreicht. Für die 2^{32}–Tests benötigt unsere Hardware rund 89 Minuten. Da die FIR–Berechnung im DSP die Simulationsgeschwindigkeit begrenzt, ist es vorteilhaft, daß die 2^{20} verschiedenen Initilisierungswerte der Register im FPGA zum gleichen Ergebnis führen müssen, da ein Einfluß der Registerinitialisierung auf den Ausgang kleiner als die Impulsantwortlänge bleiben muß und jeweils der 12. Sample getestet wird. Im DSP müssen daher (nur) 2^{12} verschiedene Sequenzen berechnet werden. Die 2^{32} Tests ergaben für $K/D = 1/6$ gleiche Ergebnisse von FPGA und DSP, d. h. es ist eine vollständige „Forced Response Stability" dieser Filterstruktur bewiesen.

Filtertyp	CLB–Aufwand (Ohne Differ.)	max. Laufzeit zw. 2 Registern	Länge der Impulsantwort
Epol	$2 \cdot B_{\max}$	1 Add.	13
ZEpol	$2{,}5 \cdot B_{\max}$	2 Add.	11
Z^2Epol	$2{,}5 \cdot B_{\max}$	3N=6 Add.	9
DA-I-5	$2{,}5 \cdot B_{\max}$	2 Add.	13
DA-II-2	$3{,}5 \cdot B_{\max}$	3 Add.	13

Tabelle 3.4.2 Vergleich der Elementarfilter nach Aufwand (Xilinx XC3000), Länge der Impulsantwort und Latenzzeiten für zweistufiges ($N = 2$) CBPC–Filter mit $K/D = 1/6$.

3.4.3 Bestimmung geeigneter Bit–Breiten in der Kaskade

Da FSF–Filter FIR–Verhalten besitzen, lassen sich die zusätzlich benötigten Bits (=ld($G_{\max}$)) für ein korrektes Verhalten durch die endliche Summe

$$G_{\max} = \sum_k |h_1[k]| \tag{3.4.2}$$

bestimmen, wobei $h_1[k]$ die Impulsantwort des Gesamtfilters beschreibt. Für CTPC, CHPC und CBPC–90°–Filter können geschlossene Lösungen angegeben werden:

$$0°, 180° \; : \; G_{\max} = D^N \tag{3.4.3}$$

$$90° \; : \; G_{\max} = (D/2)^N \tag{3.4.4}$$

Für 60° und 120° CBPC–Filter konnte die folgende obere Schranke bestimmt werden:

$$G_{\max} \leq (2 \cdot D/3)^N \tag{3.4.5}$$

Für einstufige 60° und 120° CBPC–Filter ist die Ungleichung (3.4.5) mit Gleichheit erfüllt.

Für eine erste Aufwandsabschätzung ist in der Tabelle 3.4.3 für typische Verzögerungswerte D das zugehörige Bit–Wachstum ld($G_{\max}$) angegeben. Die maximal benötigte Bit–Breite $B_{\max}$ berechnet sich somit zu

$$B_{\max} = B_{\text{in}} + \text{ld}(G_{\max}) \tag{3.4.6}$$

3.4.3.1 Bestimmung der Bit–Breite der einzelnen Register

Die am Ausgang des Dezimationsfilters gewünschte Bit–Breite beinhaltet zumeist ein sehr viel größeres Quantisierungsrauschen als das Rauschen, das bei Abschneiden (Truncation) der unteren Bits in den Elementarfiltern entsteht. Im folgenden soll dem Ansatz

Winkel	Pol K	Delay D	ld(G_{max}) bei Stufenzahl N					
			1	2	3	4	5	6
0°/180°	0/3	6	3	6	8	11	13	16
	0/6	12	4	8	11	15	18	22
	0/18	36	6	11	16	21	26	32
	0/30	60	6	12	18	24	30	36
60°/120°	1/2	6	2	4	6	7	9	10
	2/4	12	3	6	9	11	14	16
	6/12	36	5	9	13	18	22	26
	10/20	60	6	11	16	21	25	30
90°	1	6	1	2	3	4	5	6
	3	12	3	6	8	11	13	16
	9	36	5	9	13	17	21	26
	15	60	5	10	15	20	25	30

Tabelle 3.4.3 Bit–Wachstum für 1-6 stufige CBPC–Filter.

von [Hog81] folgend in jeder Stufe j eine zusätzliche Rauschkomponente σ_j^2 zugelassen werden, die in ihrer Summe über alle Stufen im Ausgangssignal eine ebenso große Rauschkomponente bewirkt wie die Quantisierung des Ausgangs selbst:

$$\sum_{j=1}^{2N} \sigma_{Tj}^2 = \sum_{j=1}^{2N} \sigma_j^2 F_j^2 \leq \sigma_{T2N+1}^2 \tag{3.4.7}$$

$$\sigma_{Tj}^2 = \frac{1}{2N} \sigma_{T2N+1}^2 \tag{3.4.8}$$

$$F_j^2 = \sum_k (h_1[k])^2 \quad j = 1, 2, \ldots 2N \tag{3.4.9}$$

wobei σ_{T2N+1}^2 das Quantisierungsrauschen durch Abschneiden am Ausgang und F_j^2 den Leistungsübertragungsfaktor der Stufe j zum Ausgang beschreibt.

Somit ergibt sich für die in Stufe j wegzulassenden Bits B_j:

$$B_j = \left\lfloor 0{,}5 \log_2 \left(F_j^{-2} \cdot \frac{6}{N} \cdot \sigma_{T2N+1}^2 \right) \right\rfloor \tag{3.4.10}$$

$$\sigma_{Tj}^2\big|_{j=2N+1} = \frac{1}{12} 2^{2B_j} = \frac{1}{12} 2^{2(B_{\text{in}} - B_{\text{out}} + \text{ld}(G_{\max}))} \tag{3.4.11}$$

Der Rechenaufwand in der Berechnung von Gleichung (3.4.10) besteht nun in der Bestimmung der F_j^2. Hierbei ist es nützlich, sich den Modulationssatz der z–Transformation

$$(-1)^n x[n] \quad \overset{z}{\longleftrightarrow} \quad X(-z) \tag{3.4.12}$$

in Erinnerung zu rufen, da durch Modulation mit $(-1)^n$ Pole $z'_{\infty i} \rightarrow -z_{\infty i}$ und Nullstellen $z'_{0i} \rightarrow -z_{0i}$ übergehen. D. h.

$$(-1)^n h_{0°}[n] = h_{180°}[n] \tag{3.4.13}$$
$$(-1)^n h_{120°}[n] = h_{60°}[n] \tag{3.4.14}$$

Für die Leistungsübertragungsfaktoren von CHPC und CTPC ergeben sich ebenso wie für 60° und 120° gleiche Ergebnisse.
Mit Hilfe der binomischen Reihe für negative Exponenten [BS84, S. 32]

$$(1-x)^{-m} = \sum_{i=0}^{\infty} \binom{m+i-1}{i} x^i \tag{3.4.15}$$

und Summation aller z^{-k} mit dem gleichen Exponenten erhält man bei 0° und 180° für die Kaskade mit Z²Epol-Elementarfiltern

$$H_j(z) = \frac{\left(1-z^{-D}\right)^N}{\left(1-\cos(2\pi K/D)z^{-1}\right)^{N-j+1}} = \sum_{k=0}^{(D-1)N+j-1} \underbrace{\sum_{l=0}^{\lfloor k/D \rfloor} (-1)^l \binom{N}{l} \binom{N-j+k-Dl}{k-Dl} \cos^k(2\pi K/D)}_{h_j(k)} z^{-k}$$
$$j = 1, 2 \ldots N \tag{3.4.16}$$

$$H_j(z) = \left(1-z^{-D}\right)^{N+1-j} = \sum_{k=0}^{2N+1-j} (-1)^k \binom{2N+1-j}{k} z^{-kD} \tag{3.4.17}$$
$$j = N+1, N+2 \ldots 2N \tag{3.4.18}$$

Für CBPC-Filter mit Z²Epol-Elementarfiltern ergibt sich

$$H_j(z) = \frac{\left(1-z^{-D}\right)^N}{\left(1-2\cos(2\pi K/D)z^{-1}+z^{-2}\right)^{N-j+1}} \tag{3.4.19}$$

$$= \sum_{l=0}^{N} (-1)^l \binom{N}{l} z^{-Dl} \sum_{w=0}^{\infty} \binom{N-j+w}{w} \underbrace{\left(2\cos(2\pi K/D) - z^{-1}\right)^w}_{\sum_{v=0}^{w} \binom{w}{v} (-1)^v (2\cos(2\pi K/D))^{w-v} z^v} z^{-w}$$
$$j = 1, 2 \ldots N \tag{3.4.20}$$

Mit der Summation aller $k = Dl + w + v$ mit demselben Exponenten z^{-k} folgt letztlich nach einigen Umformungen

$$H_j(z) = \sum_{k=0}^{(D-1)N+j-1} \underbrace{\sum_{l=0}^{\lfloor k/D \rfloor} \binom{N}{l} \sum_{v=0}^{\lfloor (k-Dl)/2 \rfloor} (-1)^{l+v} \binom{N-j+k-Dl-v}{k-Dl-v} (2\cos(2\pi K/D))^{k-Dl-2v} \binom{k-Dl-v}{v} z^{-k}}_{h_j(k)}$$

$$j = 1, 2 \ldots N \tag{3.4.21}$$

$$H_j(z) = \text{siehe Gleichung (3.4.17)} \tag{3.4.22}$$

$$j = N+1, N+2 \ldots 2N \tag{3.4.23}$$

Die Gleichungen (3.4.16) und (3.4.21) beschreiben die Z²Epol–Lösung mit minimaler Verzögerung. Für die praktische Realisierung empfiehlt es sich aber, möglichst nach jeder arithmetischen Operation im Elementarfilter und in den Differenzierern ein Register vorzusehen. Dies ist in den anderen Elementarfiltern Epol, ZEpol, DA-I und DA-II vorhanden. Die Latenzzeit der Filter wird dann unabhängig von der gewählten Stufenzahl, siehe Tabelle 3.4.2. Bei ZEpol und DA-I ergibt sich eine zusätzliche Verzögerung von z^{-D} und für Epol und DA-II eine zusätzliche Verzögerung von z^{-2D} in der Impulsantwort. Die Leistungsübertragungsfaktoren $F_j^2 = \sum |h_j[k]|^2$ der fünf Realisierungen sind jedoch gleich.

Es erscheint allerdings sinnvoller, anstatt die Gleichung (3.4.21) zu lösen, zur Berechnung der $h_j[k]$ eine bezüglich der Impulsantwort inhärente Umformung der CBPC–Struktur in die Grundform vorzunehmen und die F_j^2 durch fortgesetzte Faltung zu berechnen, siehe Bild 3.4.7.

Benutzt wird die Tatsache, daß ein EF/DIF–Paar eine endliche Impulsantwort mit einer Länge kleiner D besitzt. Die Faltung von $g_N[k]$ mit sich selbst ergibt $g_{N-1}[k]$ mit einer Länge kleiner $2D$. Die Faltung von $g_{N-1}[k]$ mit $g_N[k]$ ergibt $g_{N-2}[k]$ usw.

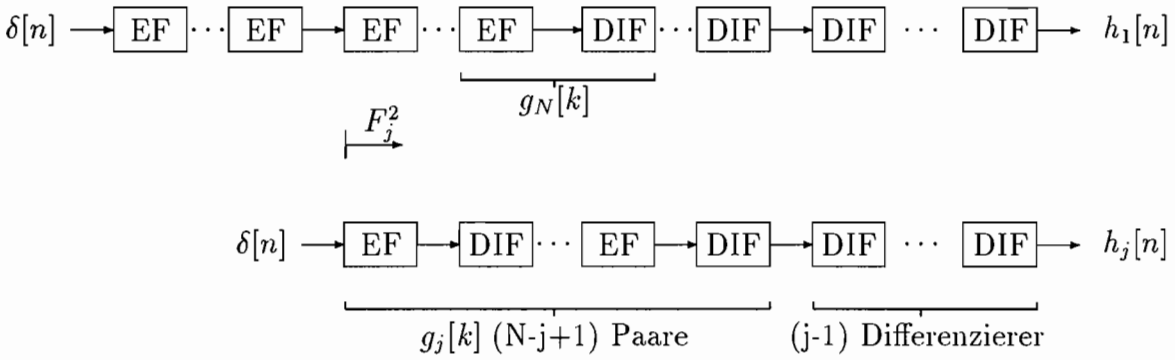


Bild 3.4.7 Umstrukturierung zur komfortablen Berechnung der Leistungsübertragungsfaktoren F_j^2.

$$g_N[k] = \begin{cases} \frac{\sin(2\pi K/D(k+1))}{\sin(2\pi K/D)} & 0 \leq k \leq D-1 \quad \text{für} \quad 60°, 90°, 120° \\ 0 & \text{sonst} \end{cases} \tag{3.4.24}$$

bzw.

$$g_N[k] = \begin{cases} \cos(2\pi K/D)^k & 0 \leq k \leq D-1 \quad \text{für} \quad 0°, 180° \\ 0 & \text{sonst} \end{cases} \tag{3.4.25}$$

$$g_{N-1}[k] = g_N[k] * g_N[k] \tag{3.4.26}$$

$$g_{N-2}[k] = g_{N-1}[k] * g_N[k] \qquad \text{usw.} \tag{3.4.27}$$

Nachdem die $g_j[k]$ alle berechnet sind, muß nun noch die Faltung mit den $j-1$ Differenzierern zum Ausgang berechnet werden. Für diese Differenzierer ergibt sich im z–Bereich $(1-z^{-D})^{j-1}$, die mit Hilfe des Binomialkoeffizienten $h_j[k]$ im Zeitbereich geschlossen angeben werden können. Es folgt letztlich:

$$h_j[k] = \sum_{i=0}^{j-1} (-1)^i \binom{j-1}{i} g_j[k-iD] \qquad j = 1, 2 \ldots N \tag{3.4.28}$$

Bei Vergleich der Gleichungen (3.4.28) und (3.4.21) wird deutlich, daß sich so die Berechnung der $h_j[k]$ und damit der Leistungsübertragungsfaktoren F_j^2 wesentlich vereinfacht.

3.4.4 Entwurfsbeispiel

Für den in Abschnitt 3.3.1 entworfenen ΣΔ–Analogteil wurde in [All92] ein passender Digitalteil konzipiert und realisiert. Die Erkenntnisse über die Stabilität bei der Umstrukturierung von CBPC–Filtern lagen zum damaligen Zeitpunkt noch nicht vor, so daß eine FSF in konventioneller Abfolge zur Anwendung kam. Das realisierte Überabtastverhältnis von $f_a/f_0 = 12$ bedingt zwei mögliche Realisierungen, welche die geforderte Auflösung erfüllen:

a) Zweistufiges FSF–Filter mit zwei parallelen Elementarfiltern, siehe Bild 3.4.8

b) Kombination aus Halbbandfilter und vierstufigen FSF mit je einem Elementarfilter mit Koeffizienten 0,5

Dabei wird sich zeigen, daß die nötigen Speicherelemente ($D = 60$) außerhalb des FPGAs realisiert werden müssen, da die Xilinx 3000 keine optimale Realisierung „größerer“ Speicher zulassen.

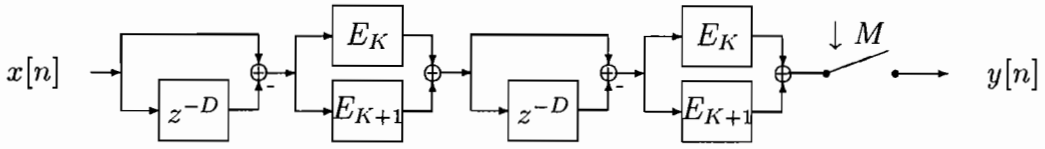


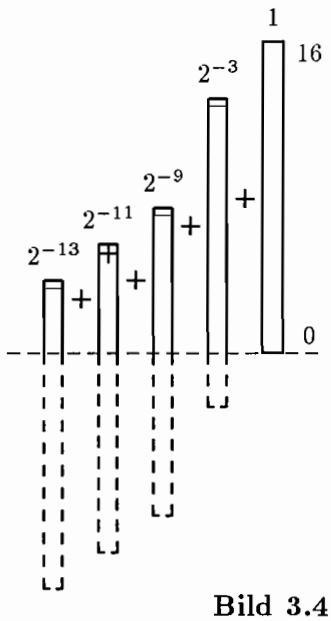
Bild 3.4.8 Dezimator als Kaskade aus 2 Frequenzabtastfiltern.

3.4.4.1 Realisierung mit parallelen Elementarfiltern

Die Untersuchungen in [All92] ergaben, daß bei der Lösung **a)** die cos-Multiplikationen bei 18 Bit Breite in 5 CSD-Koeffizienten genügend genau realisiert werden können. Der Aufwand für die Multiplizierer läßt sich erheblich reduzieren, wenn man einen kleinen Rundungsfehler akzeptiert und nur diejenigen MSB realisiert, die unmittelbar einen Beitrag zum Multiplikationsergebnis liefern, siehe Bild 3.4.9. Für den Realisierungsaufwand des Multiplizierers ergibt sich

$$CLB = B(Op-1) + Op - 2 - \sum_{n=2}^{Op-1} \mathrm{ld}|t_i|2^i \tag{3.4.29}$$

bei Realisierung des 1 Bit Ripple Carry Addierers in einem CLB. Es wird deutlich, daß der Aufwand nun nicht mehr allein, wie bei der normalen CSD-Multiplikation, von der Anzahl der CSD-Koeffizienten (Op) abhängig ist, sondern auch von dem Wert des jeweiligen Koeffizienten t_n.



$$\begin{aligned} D = 210 \quad &; \quad K = 17 \\ \cos(2\pi K/N) &= 0,873413085 \\ &\cong 1,0 - 2^{-3} - 2^{-9} + 2^{-11} - 2^{-13} \\ &= 1,00\bar{1}000000\bar{1}010\bar{1} \\ &\hat{=} \ 17,64 \text{ eff. Bit genau} \end{aligned}$$

Bild 3.4.9 Realisierung der Multiplizierer im CSD-System.

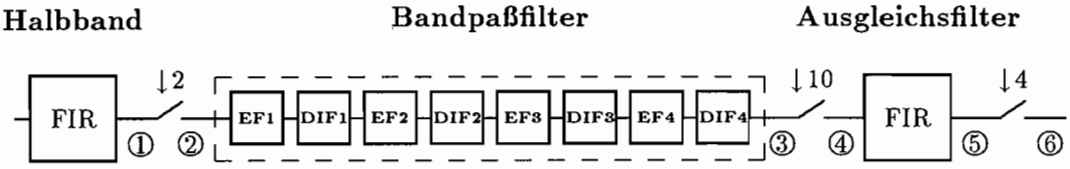


Bild 3.4.10 Blockschaltbild des realisierten Bandpaßsystems.

Insgesamt ergibt sich der Aufwand für die Filterstruktur zu:

Struktur	Aufwand in CLBs
2 Addierer (17 Bit)	34
1 Differenzierer (11 Bit)	22
1 Differenzierer (14 Bit)	28
2 Elementarfilter (17 Bit, mit C_1)	2(52+48)=200
2 Elementarfilter (17 Bit, mit C_2)	2(52+52)=208
Gesamt	492

Hierzu müßten noch einige CLBs für das Steuerwerk gerechnet werden, so daß der Gesamtaufwand mehr als 500 CLBs beträgt. Xilinx FPGAs mit solch einer Komplexität sind zur Zeit noch nicht erhältlich, daher ist diese Filterstruktur nicht in einem FPGA zu realisieren.

3.4.4.2 Realisierung mit Halbbandfilter und Elementarfiltern

Für die Realisierung der zweiten Alternative ergeben sich günstigere Ergebnisse. Als Halbbandfilter wurde ein elfstufiges Filter [CR83, S. 231,F5] gewählt. Für das Filter ergibt sich ein maximales Bit–Wachstum von $\lceil \sum |h_i| \rceil$ =9 Bit. Zur Verringerung der Laufzeit

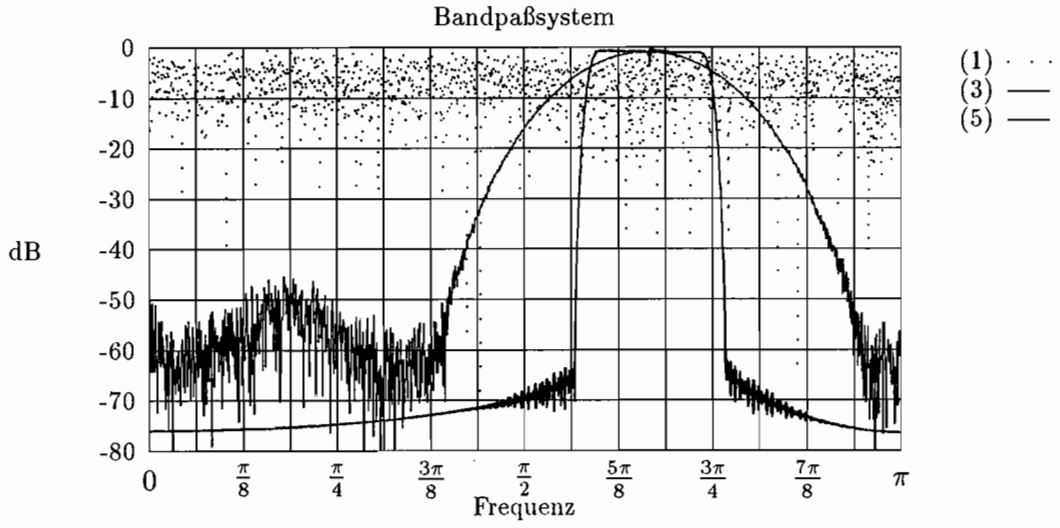


Bild 3.4.11 Übertragungsfunktionen des realisierten Bandpaßsystems entsprechend Bild 3.4.10 bei Anregung mit einem Wobbelsignal. (1) Spektrum am Ausgang des Halbbandfilters. (3) Spektrum am Ausgang des FSF. (5) Spektrum am Ausgang des FIR–Ausgleichfilters.

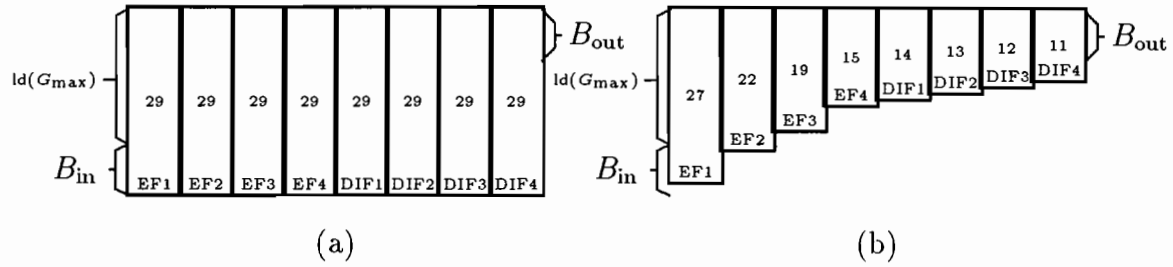


Bild 3.4.12 (a) EFilter ohne Detailentwurf der Bit–Breiten. (b) EFilter mit Detailentwurf der Bit–Breiten.

des Filters wurde das Filter in ein Schieberegister mit Tabelle umgesetzt. So konnte das Halbbandfilter mit 42 CLBs realisiert werden. Zusammen mit einem hybriden 24 Bit Akkumulator–Synthesizer (siehe Abschnit 2.3.2.1) mit Frequenzvervielfachung durch eine PLL mit Teiler wurden 73 CLBs benötigt. Dieser Schaltungsteil wurde in einem XC3042 realisiert.

Das Bandpaßfilter wurde aus Elementarfiltern mit cos–Koeffizienten von 0,5 realisiert und benötigt daher nur $3 \cdot b$ CLBs je b Bit breitem Elementarfilter. Günstig ist weiterhin, daß, da Elementarfilter und Differenzierer sich abwechseln, das vorhandene Bitwachstum nur innerhalb einer Stufe berücksichtigt werden muß und daher wesentlich geringer ist als bei der CBPC–Struktur, siehe Bild 3.4.12. Es werden nur zwei 14 und zwei 15 Bit breite Elementarfilter benötigt. Insgesamt berechnet sich der Aufwand folgendermaßen:

Struktur	Aufwand in CLBs
3 Differenzierer (10 Bit)	60
1 Differenzierer (13 Bit)	26
2 Elementarfilter (14 Bit)	84
2 Elementarfilter (15 Bit)	90
Gesamt	260

Insgesamt ergibt sich somit ein Aufwand 260+42=302 CLBs für das Dezimationsfilter. Gegenüber den 492 CLBs bei der parallelen zweistufigen Realisierung erscheint somit die vierstufige Lösung mit Halbbandfilter günstiger und wurde deshalb in [All92] realisiert. Der Ausgleich der si^4–förmigen Spektrums erfolgt in einer abschließenden FIR–Stufe, siehe Bild 3.4.11. Nach der abschließenden FIR–Filterung ist eine weitere Reduktion um $M = 4$ möglich.

3.4.4.3 Realisierung mit CBPC–Filtern

Ausblickend soll betrachtet werden, welche Einsparungen durch die vorgeschlagene CBPC–Filterstruktur möglich sind.

Der Entwurf einer Kaskade mit geeigneten Bit–Breiten sei an dem vierstufigen CBPC–Filter des letzten Abschnitts bei Nullstellenwinkel 60° gezeigt. Weitere Daten des Filters sind: $B_{\mathrm{in}} = 8$ Bit, $B_{\mathrm{out}} = 12$ Bit, $D = 60$, $M = 10$.

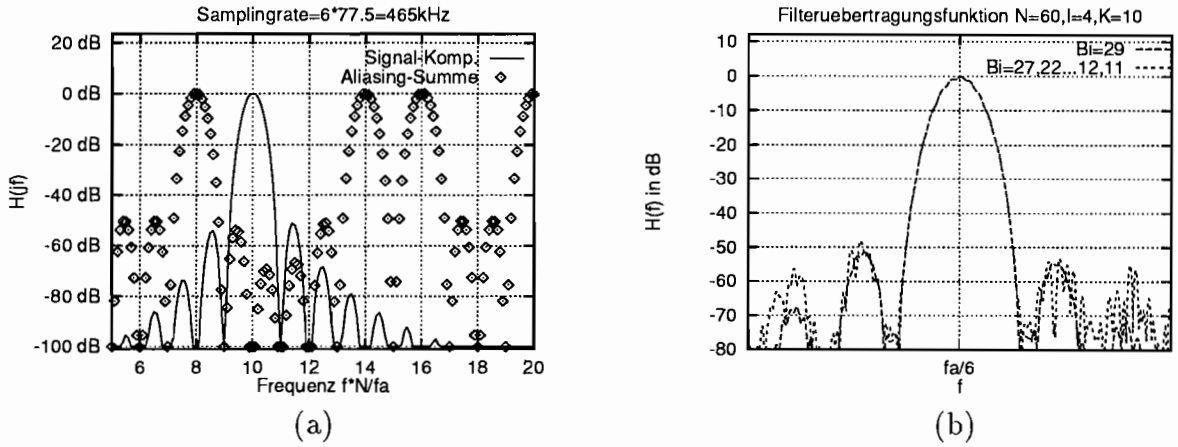


Bild 3.4.13 (a) Berechnetes Summenspektrum aller Aliasingkomponenten ◇ und die Signalkomponenten —— des FSF–Filters nach Gleichung 3.4.1 und Reduktion um $\downarrow M = 10$. (b) Simuliertes Spektrum des Bandpasses durch Fourier–Transformation Antwort des Filters auf ein Wobbel–Signal. —— Alle Stufen 29 Bit. - - - Elementarfilter bei Bit–Breiten: 1. 27 Bit, 2. 22 Bit,3. 19 Bit 4. 15 Bit; Comb–Stufen :1. 14 Bit,2. 13 Bit, 3. 12 Bit, 4. 11 Bit.

Aus der Tabelle 3.4.3 ergibt sich für $K/D = 10/60$ (4 stufig) ein Bit–Wachstum von $\mathrm{ld}(G_{\max}) = 21$ Bit. Bild 3.4.12 zeigt die Ergebnisse des Detailentwurfs der Bit–Breiten der einzelnen Stufen. Für eine Abtastung mit $6 \cdot 77,5$ kHz und eine Durchlaßbreite von 5 kHz ergibt sich mit Hilfe des objektorientierten C++ Simulators **FiSK/AL** [All92] (mit und ohne Detailentwurf der Bit–Breiten bei Anregung mit einem Wobbelsignal und der im Bildtext angegebenen Bit–Breiten) das Spektrum aus Bild 3.4.13(b). In[MB93a, Anhang A] ist die Beschreibung der Filterstruktur aus Bild 3.4.10 angegeben. In Bild 3.4.13(a) ist das mit Hilfe der Gleichung (3.4.1) für $z = e^{j2\pi f/f_a}$ berechnete Spektrum des BP–Filters nach der Reduktion um den Faktor $\downarrow M = 10$ gezeigt. Es ergibt sich eine Sperrdämpfung von 53,8 dB. Ist Aliasing im Übergangsbereich erlaubt, so wird eine Sperrdämpfung von 74,9 dB erreicht, was für einen 8 bzw. 12 Bit A/D–Wandler ausreichend ist.

Beim Vergleich des Aufwands der Realisierungen in einem FPGA, müssen die XC3000 und XC4000 Serie unterschieden werden, da in der XC4000 Serie Speicherelemente effizienter realisiert werden können, siehe Tabelle 3.4.4. Deutlich wird, daß eine erhebliche Aufwandsreduktion erreicht wird. Bei der XC3000–Serie resultiert ein Aufwand von

Typ	XC3000	XC4000
FSF	3712	342
EPOL	580	232
Detail EPOL	316	183

Tabelle 3.4.4 Realisierungsaufwand des Entwurfsbeispiels (ohne Steuerwerk für den Gesamtablauf) in CLB.

8,5% und bei der XC4000-Serie von 53% verglichen mit dem konventionellen Entwurf des FSF-Filters.

Der Vergleich dieser Ergebnisse mit dem realisierten Entwurf aus Abschnitt 3.4.4.1 zeigt, daß die dort beschriebenen 260 CLB noch weniger sind als die hier nötigen 316 CLBs. Dabei muß aber berücksichtigt werden, daß bei der CBPC-Realisierung im Gegensatz zu der Realisierung in Abschnitt 3.4.4.1 keine externe RAM/FIFO-Bausteine benötigt werden.

Kapitel 4

Realisierung von universellen Demodulatoren

Wird die Einteilung der Empfängerprinzipien in [MG92, Q4] aufgegriffen, so wird deutlich, daß als Universalempfänger nur Überlagerungsempfänger geeignet sind. Weiterhin unterscheidet man die Lage der Zwischenfrequenz. Beim **Homodynempfänger** (Null–ZF–Empfänger) ist die Mischfrequenz gleich der Signalfrequenz (Trägerfrequenz). Dieses Prinzip zeichnet sich durch geringen Aufwand und günstige monolithische Realisierungen aus. Der **Hederodynempfänger** (Einfachsuperhet, Doppelsuperhet) verwendet zusätzliche Zwischenfrequenzbereiche, so daß die Hauptselektion verbessert und vereinfacht wird. Im folgenden werden verschiedene Demodulationsprinzipien durch Aufwandsabschätzungen, Rechnersimulationen und Schaltungsaufbauten miteinander verglichen. Bei dem eigentlichen Demodulationsprinzip werden kohärente (synchrone) und inkohärente (asynchrone) Prinzipien unterschieden. Bild 4.0.1 zeigt die Übersicht einer Empfängereinteilung.

Von den universellen Empfängern wurde bisher der kohärente PLL–Empfänger [Bel89] bei Realisierung mit Mikrocontroller (MC) und der inkohärente Quadratur(abtast)empfänger [Wie88] bei Realisierung mit digitalem Signalprozessor (DSP) in Bezug auf das DCF77 Signal ausführlich untersucht. Im folgenden Kapitel werden spezielle Aspekte bei der Realisierung universeller Demodulationsprinzipien mit FPGAs untersucht.

In Abschnitt 4.2 werden FPGA Realisierungen des universellen CORDIC–Demodulators beschrieben. In Abschnitt 4.3 wird dann auf besondere Möglichkeiten der gedächtnisbehafteten Demodulation von FM–Signalen eingegangen. Im Abschnitt 4.4 wird die Realisierung kohärenter Verfahren (PLL, Costas–Loop) erörtert. Im Schlußabschnitt dieses Kapitels werden realisierte Empfängerkonzepte kurz erläutert und eine vergleichende Messung der Universalempfänger vorgestellt.

4.1 Inkohärente Abtastempfänger für Bandpaßsignale

Zunächst werden Randbedingungen einer FPGA–Implementierung des klassischen Quadraturüberlagerungsprinzips bei Abtastung oberhalb der Nyquistfrequenz diskutiert.

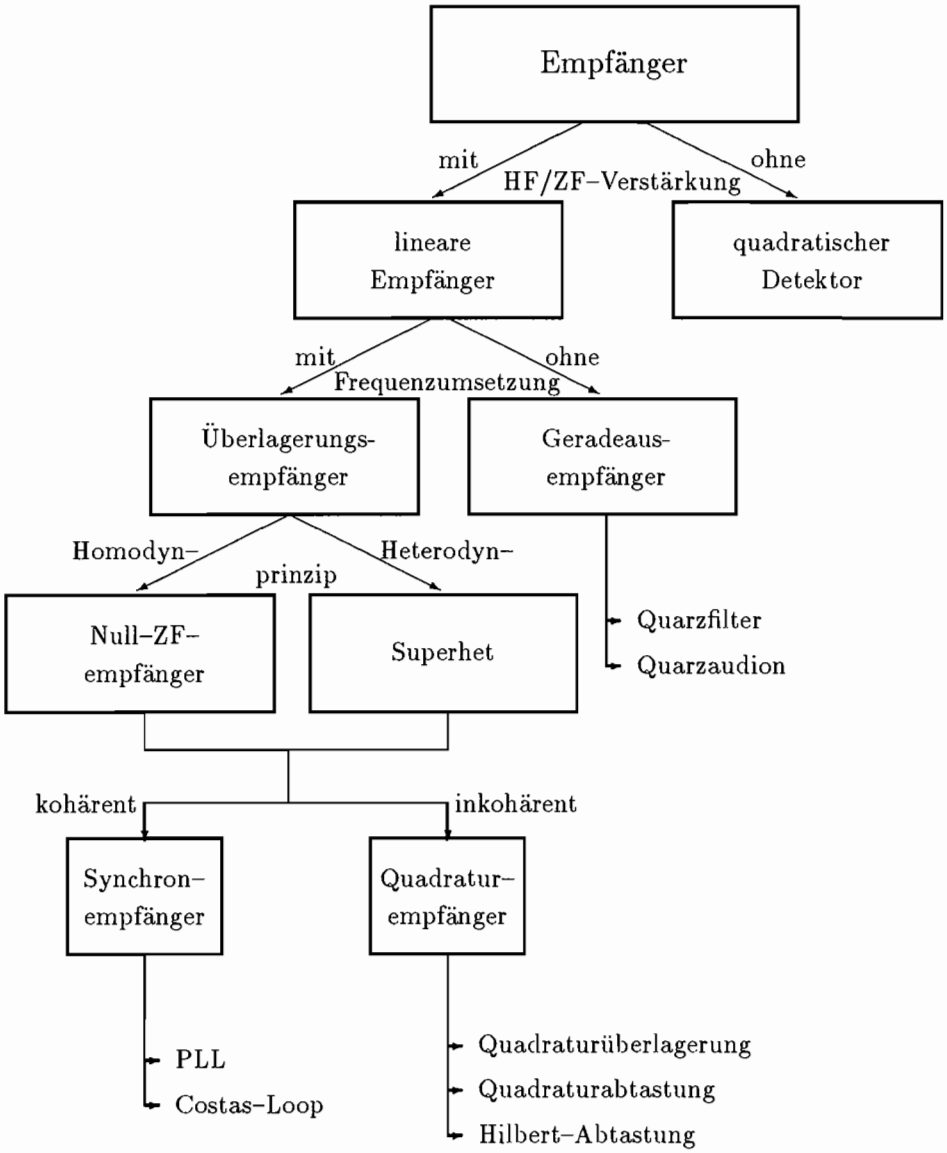


Bild 4.0.1 Einteilung der Empfänger angelehnt an [MG92, Q4].

Danach werden speziell für Bandpaßsignale günstige Abtastempfänger (Quadraturabtastung Abschnitt 4.1.1), und die neuartige Hilbert–Abtastung Abschnitt 4.1.4) miteinander verglichen. Bevor auf die Unterschiede zwischen Quadraturabtastempfänger und Hilbert–Abtastempfänger eingegangen wird, werden in Abschnitt 4.1.2 für (Ganzzahl–)Bandpaßabtastung (Integer–Band–Sampling) die prinzipielle Auswirkungen der Unterabtastung verdeutlicht.

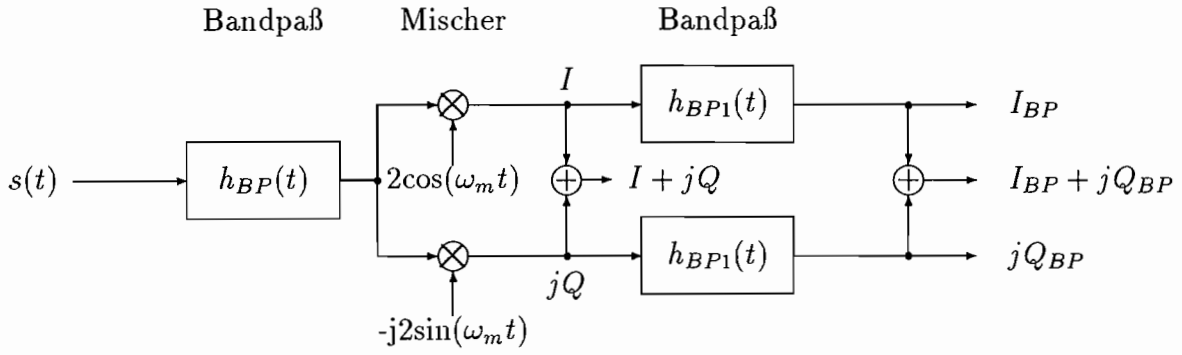


Bild 4.1.1 Analoge Realisierung zur Erzeugung der I– und Q–Phase (Quadraturüberlagerungsempfänger).

4.1.1 FPGA Implementierung von Quadraturüberlagerungsempfängern

Bild 4.1.1 zeigt die analoge Realisierung der I/Q–Erzeugung durch Mischen mit den Signalen $2\cos(2\pi f_m t)$ und $-j2\sin(2\pi f_m t)$. Bei dieser Realisierung muß im ZF–Bereich $f_z = f_0 - f_m$ eine Bandpaß–Filterung durchgeführt werden. Zu beachten ist, daß das resultierende Spektrum keine Anteile bei negativen Frequenzen besitzt. Um die Anforderungen an den Bandpaß möglichst gering zu halten, ist man bestrebt, f_m möglichst nahe an f_0 zu legen.

Wird die Mischung digital ausgeführt, so ist eine **Null–ZF–Umsetzung** (Homodynempfang) möglich, d. h. $f_m = f_0$, was bei einer *analogen* Realisierung wegen der auftretenden Eigenstörung zu erheblichen Dynamikproblemen bei einer anschließenden A/D–Wandlung führt [Wie92b]. Die Null–ZF–Umsetzung ist im Blockschaltbild 4.1.2 dargestellt. Bild 4.1.3 zeigt die zugehörigen Spektren. Bild 4.1.3(a) zeigt das tiefpaßgefilterte Bandpaßsignal am Eingang des Empfängers. Durch die Abtastung entsteht das zeitdiskrete Spektrum $S(e^{j\omega})$. Nach der Multiplikation mit dem cos–Signal ($\circ\!\!-\!\!\bullet$ $S_{2\cos}(e^{j\omega})$) bzw. dem sin–Signal ($\circ\!\!-\!\!\bullet$ $S_{-j2\sin}(e^{j\omega})$) entstehen die Inphasekomponente $I(e^{j\omega})$ und die Quadraturkomponente $jQ(e^{j\omega})$, siehe Bild 4.1.3(e) und (f). Diese zwei zeitdiskreten

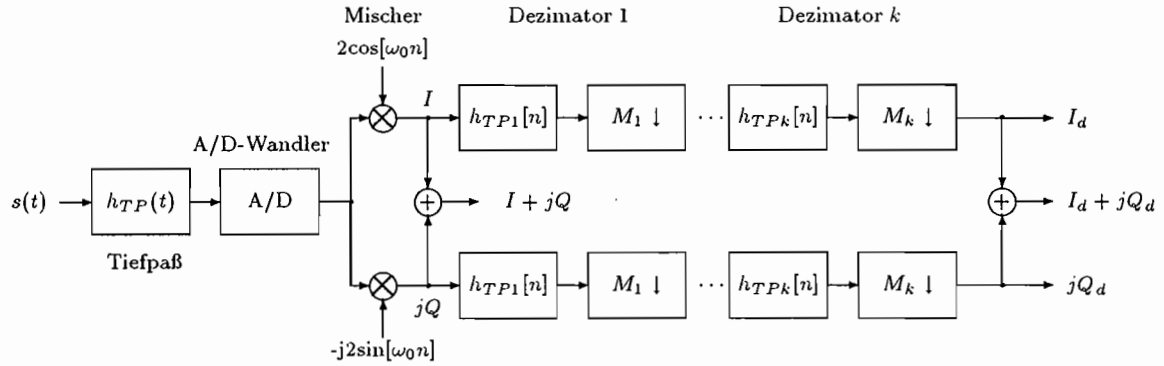


Bild 4.1.2 Null–ZF–Umsetzung realisiert mit einem FPGA. Nach Nyquistfilterung und Abtastung erfolgt die digitale sin/cos–Multiplikation. Es folgt die Tiefpaß–Filterung ($h_{TPi}[n]$) und Abtastratenreduktion $\downarrow M_i$ in k Stufen. Abschließend erfolgt die Zusammenfassung zu einem komplexen Signal $I_d + jQ_d$.

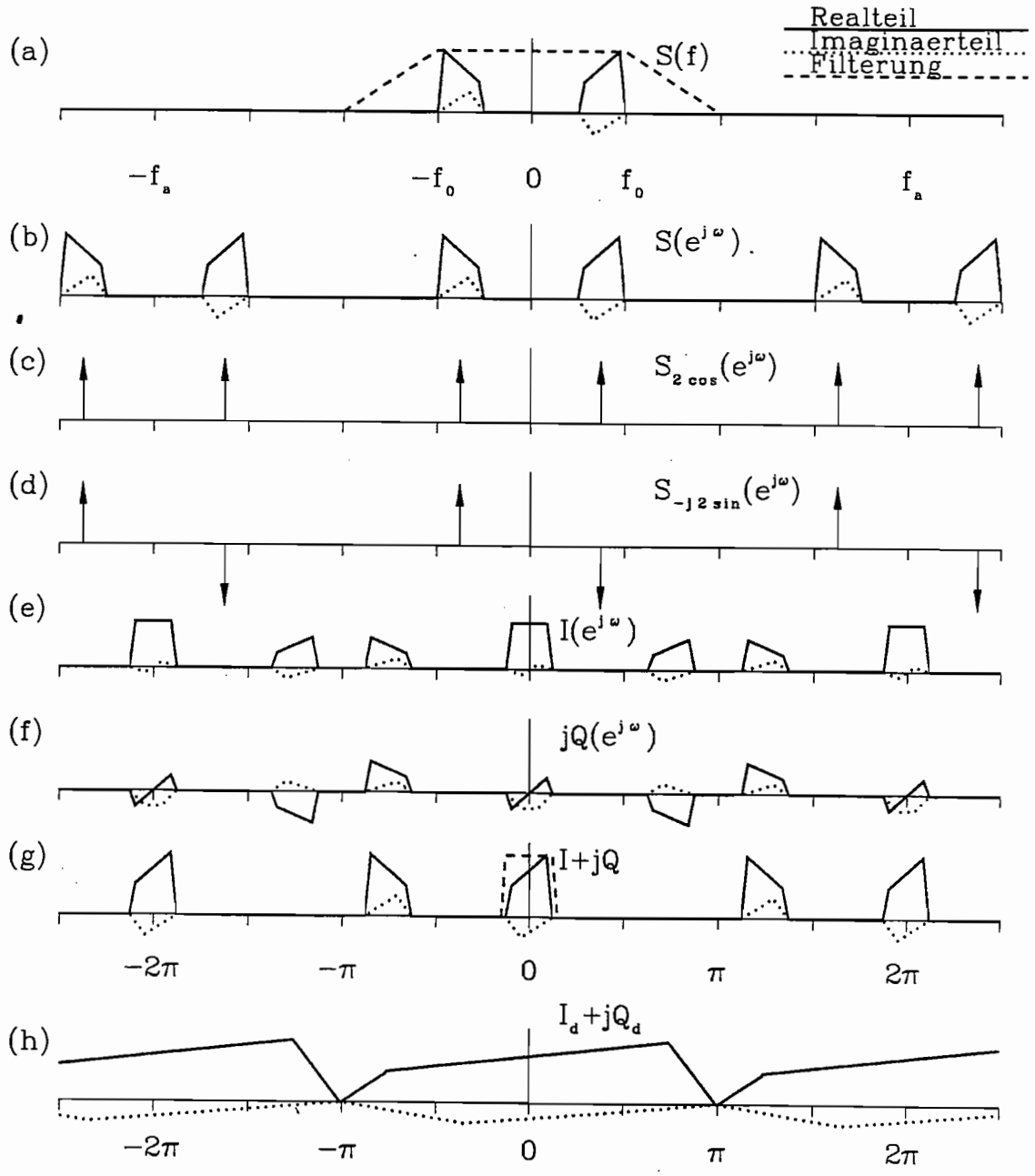


Bild 4.1.3 Spektrale Darstellung der Null–ZF–Umsetzung. Der Abtastfrequenz f_a im Analogbereich von $S(f)$ entspricht 2π im Digitalbereich $S(e^{j\omega})$.

Signale werden zu einem komplexen Signal $I+jQ$ zusammengefaßt, siehe 4.1.3(g). Abschließend muß nur noch eine Tiefpaßfilterung ausgeführt werden. Es wird deutlich, daß bei der Null–ZF–Umsetzung lediglich eine Tiefpaß–Filterung und keine Bandpaß–Filterung nötig ist. Zu beachten ist allerdings bei Null–ZF–Umsetzung, daß das Eingangssignal mit $f_a \geq 4f_0$ abgetastet werden muß, damit es durch den Anteil bei $\pm\pi \cong \pm f_a/2$ zu

keinen Störungen kommt, siehe Bild 4.1.3(g). In Bild 4.1.3(h) ist das Spektrum des komplexen Zeitsignals $I_d + jQ_d$ nach der Dezimation dargestellt. Es ist aus Bild 4.1.3(h) zu erkennen, daß das Basisspektrum ($|\omega| < \pi$) bei Null–ZF–Umsetzung im Gegensatz zu Bild 1.1.2(c) auch „negative" Frequenzanteile enthält. Dennoch ist es *analytisch*, da es komplex differenziebar ist [BS84, S. 518].

Die gesamte Null–ZF–Umsetzung läßt sich vorteilhaft mit einem PGA realisieren, da hier Mischung und Dezimation (Tiefpaß–Filterung mit Abtastratenreduktion [CR83, S. 31-35]) in einem Baustein realisiert werden können. Vorteilhaft ist es, den Tiefpaß als „Multistage"–Realisierung (mehrere Tiefpässe und Reduzierer mit jeweils geringerer Reduktion) auszuführen [CR83, S. 196-204], da so bei geringerer Anforderung an die Tiefpässe schmale Übergangsbandbreiten zwischen Durchlaß– und Sperrbereich erreicht werden. In [AW91] wurde dieses Null–ZF–Prinzip für Empfangsfrequenzen 50-150 kHz in einem FPGA realisiert. Am Eingang des Empfängers sind eine breitbandige Ferrit–Antenne, Verstärker und eine digital einstellbare AGC mit MDAC vorhanden. Als Aliasing–Filter ist ein Cauer–Tiefpaß 7. Ordnung bei einem Übergangsbereich von 150-200 kHz vorhanden, der die geforderte Sperrdämpfung des 8 Bit A/D–Wandler von 48 dB bei den Spiegelfrequenz der tiefsten Signalfrequenz erzeugt. Ein PLL–basierter Synthesizer mit Akkumulatorreferenz liefert die achtfache Signalfrequenz (0,400-1,2 MHz), so daß die benötigten sin/cos–Koeffizienten für alle Eingangsfrequenzen in einer kleinen Tabelle Platz finden. Als sin/cos–Mischer wird *ein* 8x8 Bit vorzeichenbehafteter Multiplizierer im Multiplex betrieben [Spa76, S. 61]. Als Tiefpaßfilter wurde ein zweistufiges CIC–Filter mit Detailentwurf der Bit–Breiten, wie es im Abschnitt 3.4.3 beschrieben ist (Integrierer: 24 und 19 Bit; Differenzierer: 17 und 16 Bit), realisiert. Zwischen Integrierer und Differenzierer liegt die für CIC–Filter übliche Abtastratenreduktion um $\downarrow M = 64$. Durch die Reduktion wird die nachfolgende Signalverarbeitung stark entlastet. Ein Signalprozessor allein ist nicht in der Lage, die Aufgabe eines PGA–Bausteins entsprechend Bild 4.1.2 bei 1,2 MHz Abtastfrequenz zu bewältigen, da dem schnellsten DSP lediglich 15 MHz/1,2 MHz=12,5 Zyklen zur Bearbeitung eines Abtastwertes zur Verfügung stünden. Zum Zeitpunkt dieser Realisierung (1991) war dies der erste volldigitale Quadraturempfänger mit nur einem Baustein. Andere Lösungen [Ple90b] oder [Har92] benötigten 3 Bausteine. Die FPGA–Schaltung entsprechend Tabelle 4.1.1 konnte in einem XC3090 Baustein realisiert werden.

Schaltungsteil	CLB–Aufwand
Mischer mit sin/cos–Tabellen	74
2 CIC–Filter	168
Ablaufsteuerung und DSP–Interface	18
Frequenzerzeugung	32
Summe	292

Tabelle 4.1.1 Aufwandsbetrachtung zum Quadraturüberlagerungsempfänger mit FPGA [AW91].

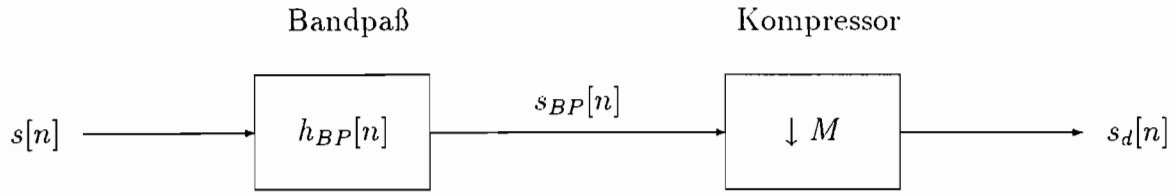


Bild 4.1.4 System zur Integer–Band–Dezimation.

4.1.2 Die Bandpaßabtastung (Integer–Band–Sampling)

Ziel der Abtastempfänger mit Unterabtastung ist es, das Bandpaß–Signal durch ein entsprechendes Tiefpaßsignal darzustellen, dessen Abtastrate in der Größe der Bandbreite der Bandpaß–Signals liegt und nicht bei der Nyquistrate des hochfrequenten Bandpaß–Signals.

Liegt das Integer–Band des unterabzutastenden Signals in dem Bereich zwischen $\omega_u = k\frac{\pi}{M}$ und $\omega_o = (k+1)\frac{\pi}{M}$ und beträgt die Bandbreite $\frac{\pi}{M}$, dann ist es möglich, die Abtastrate gegenüber der Nyquist–Rate um einen Faktor M zu reduzieren, ohne daß Aliasing oder ein anderer Informationsverlust auftritt [CR83, S. 43-47]. Aus Bild 4.1.5 wird deutlich, daß das Signal mit $f_0 = 2\Delta f$ abgetastet sein muß, damit es einerseits zu keinen Überschneidungen bei $S_I(e^{j\omega})$ kommt und andererseits die periodischen Wiederholungen der Bandbreite $\frac{\pi}{M}$ so dicht aneinanderliegen, daß die Abtastrate minimal ist. Diese stimmt mit der Abtastrate für das komplexe äquivalente Basisbandsignal überein und ist deshalb minimal.

Spektral betrachtet, entspricht die Operation $\downarrow M$ (Kompressor) aus Bild 4.1.4 einer Modulation bzw. Faltung mit einer Dirac–Stoßfolge $D_I(f)$. Das resultierende Spektrum $S_I(e^{j\omega})$ im Bereich $-\pi < \omega < \pi$ ergibt sich nach der Reduktion zu

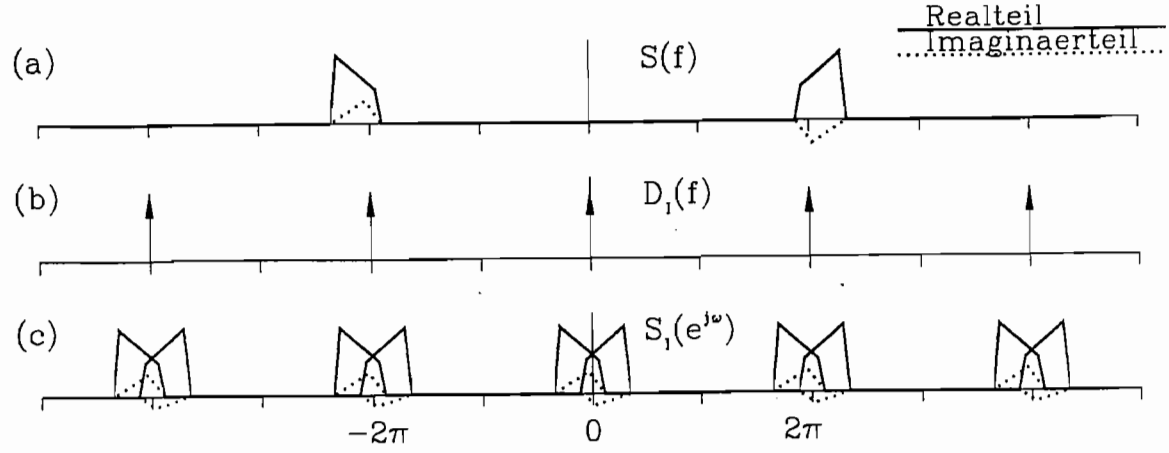


Bild 4.1.5 Spektrale Darstellung der „Integer–Band"–Verletzung.

$$S_{\mathrm{I}}(e^{j\omega}) = \frac{1}{M}\sum_{l=0}^{M-1} H_{\mathrm{BP}}\left(e^{j(\omega-2\pi l)/M}\right) S\left(e^{j(\omega-2\pi l)/M}\right) \tag{4.1.1}$$

Wird die ganzzahlige Beschränkung $\omega = k\frac{\pi}{M} \cdots (k+1)\frac{\pi}{M}$ nicht eingehalten, so kommt es, obwohl die Abtastrate wesentlich oberhalb der geforderten Rate $f_0 = 2\Delta f$ liegen kann, zu Überschneidungen im Spektrum. Dies wird in der Literatur mit „Integer–Band Violation" bezeichnet und ist in Bild 4.1.5 im Spektralbereich dargestellt. Das Eingangssignal $S(f)$ wird durch eine Dirac–Stoßfolge, deren Spektrum in Bild 4.1.5(b) dargestellt ist, unterabgetastet. Das resultierende Spektrum $S_{\mathrm{I}}(e^{j\omega})$ in Bild 4.1.5(c) zeigt die unerwünschten Überlappungen im Spektrum. Daraus folgt: „Integer–Band Violation" führt auf eine nichttriviale Bestimmung von k, M und f_0, wenn man einen universellen Empfänger realisieren möchte, der für variable Bandbreiten und Mittenfrequenzen fehlerfrei arbeiten soll.

4.1.3 Der Quadraturabtastempfänger

Der Quadraturabtastempfänger wurde in [Hil88, S. 107-125] und [Wie88, S. 87-95] ausführlich systemtheoretisch untersucht, soll aber an dieser Stelle nochmals mit Hilfe spektraler Darstellungen kurz veranschaulicht werden. Dabei soll zugleich das Verhalten bezüglich "Integer–Band Violation" noch näher untersucht werden. Die Abtastung des Bandpaß–Signals erfolgt hier, ähnlich wie bei der Quadraturüberlagerung (siehe Abschnitt 4.1.1), durch zwei um 90° oder $T_A/4$ verschobene periodische Abtastungen. Mit $f_A = 1/T_A$ ergibt sich für den I– bzw. Q–Zweig, siehe Bild 4.1.6:

$$d_{\mathrm{I}}(t) = \sum_{n=-\infty}^{\infty} T_A\delta(t-nT_A) \quad \circ\!\!-\!\!\bullet \quad D_{\mathrm{I}}(f) = \sum_{n=-\infty}^{\infty} \delta(f-nf_A) \tag{4.1.2}$$

$$d_{\mathrm{Q}}(t) = \sum_{n=-\infty}^{\infty} T_A\delta(t-nT_A+\frac{T_A}{4}) \quad \circ\!\!-\!\!\bullet \quad D_{\mathrm{Q}}(f) = \sum_{n=-\infty}^{\infty} \delta(f-nf_A)e^{j\frac{\pi}{2}n} \tag{4.1.3}$$

Diese Abtastimpulse sind mit dem Zeitsignal zu multiplizieren und werden anschließend nach einer Bandpaß–Filterung zu einem komplexen Signal zusammengefaßt, das *analytisch* oder *konjugiert komplex analytisch* ist.

$$\begin{aligned}
s_{\mathrm{I}}(t) = s(t)d_{\mathrm{I}}(t) \quad &\circ\!\!-\!\!\bullet \quad S(f) * D_{\mathrm{I}}(f) = S_{\mathrm{I}}(f) \\
S_{\mathrm{I}}(f) \quad &= \quad \frac{1}{2}\sum_{n=-\infty}^{\infty} H_T(f-f_0-nf_A) + H_T^*(-f-f_0+nf_A) \\
s_{\mathrm{Q}}(t) = s(t)d_{\mathrm{Q}}(t) \quad &\circ\!\!-\!\!\bullet \quad S(f) * D_{\mathrm{Q}}(f) = S_{\mathrm{Q}}(f) \\
S_{\mathrm{Q}}(f) \quad &= \quad \frac{1}{2}\sum_{n=-\infty}^{\infty} H_T(f-f_0-nf_A)e^{j\frac{\pi}{2}n} + H_T^*(-f-f_0+nf_A)e^{j\frac{\pi}{2}n}
\end{aligned}$$

Die Zusammenfassung zu einem komplexen Signal führt auf

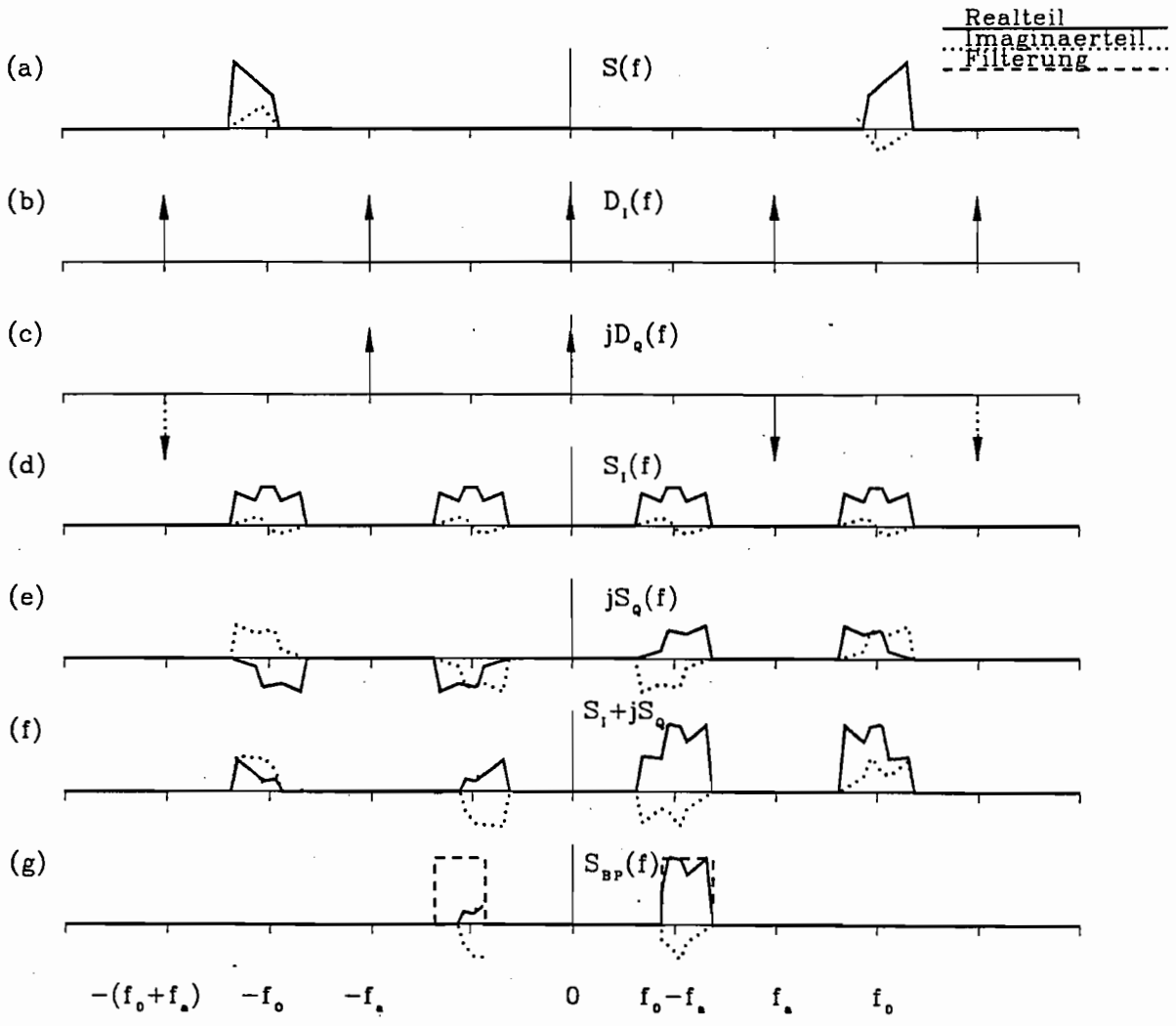


Bild 4.1.6 Spektrale Darstellung der Quadraturabtastung, wenn die ganzzahlige Bandaufteilung nicht eingehalten wurde.

$$
\begin{aligned}
S_C(f) &= S_{\mathrm{I}}(f) \pm j S_{\mathrm{Q}}(f) \qquad (4.1.4)\\
&= \frac{1}{2} \sum_{n=-\infty}^{\infty} H_T(f - f_0 - n f_A) \left[1 \pm e^{j\frac{\pi}{2}(n+1)}\right] + H_T^*(-f - f_0 + n f_A) \left[1 \pm e^{j\frac{\pi}{2}(n+1)}\right]
\end{aligned}
$$

Es kann nun das obere Seitenband bei $f_0 + f_A$ oder das untere Seitenband bei $f_0 - f_A$ zur Weiterverarbeitung benutzt werden. Außerdem sind die Kombinationen $s_{\mathrm{I}}(t) \pm j s_{\mathrm{Q}}(t)$ möglich, so daß sich 4 Möglichkeiten ergeben, siehe Tabelle 4.1.2.

	$f_0 + f_a$	$f_0 - f_a$
$s_I(t) + s_Q(t)$	$S_C(f)$ konjugiert komplex analytisch	$S_C(f)$ analytisch
$s_I(t) - s_Q(t)$	$S_C(f)$ analytisch	$S_C(f)$ konjugiert komplex analytisch

Tabelle 4.1.2 Kombinationsmöglichkeit zur Bildung analytischer Signale bei der Quadraturabtastung.

In Bild 4.1.6 wurde die Kombination aus dem unteren Seitenband ($f_0 - f_a$) und der Summe der Zweige ($s_I(t) + js_Q(t)$) ausgewählt. Es stellt sich abschließend die Frage, ob bei Unterabtastung des Signals selbst eine „Integer–Band Violation" entstehen kann. Dies ist in Bild 4.1.6 veranschaulicht. Bild 4.1.6(a) zeigt das Bandpaßsignal am Eingang des Empfängers. Nach der Inphase–Abtastung (o—• $D_I(f)$) bzw. Quadratur–Abtastung (o—• $D_Q(f)$) entstehen die Inphasekomponente $S_I(f)$ und die Quadraturkomponente $jS_Q(f)$. Diese zwei reellen Signale werden zu einem komplexen Signal $S_I + jS_Q$ zusammengefaßt, siehe 4.1.6(f). Hier wird bereits die Verfälschung des Bandpaß–Signals deutlich. Entsprechend Bild 4.1.1 muß abschließend noch die Bandpaßfilterung ausgeführt werden. Deutlich wird bei dieser Bandaufteilung, daß, wie bei der „Integer–Band"–Dezimation, eine „Integer–Band–Violation" bei direkter Unterabtastung auftritt.

Liegt bei der Quadraturabtastung nach der ersten Reduktionsstufe ein korrektes analytisches Signal vor, so ist, unter Beachtung des Abtasttheorems für Bandbaß–Signale, die zweite Dezimationsstufe aliasingfrei. Als Fazit können wir festhalten: Der Quadraturabtastempfänger eignet sich sehr gut zur Erzeugung eines analytischen Bandpaß–Signals. Schwierig bleibt für einen universellen Empfänger die Wahl der Verhältnisse M, k und f_A, mit denen es zu keiner „Integer–Band"–Verletzung kommt.

4.1.4 Der Hilbert–Abtastempfänger

In Abschnitt 1.1.1 wurde gezeigt, daß der ideale Hilbert–Transformator nur näherungsweise realisiert werden kann. Für ein schmalbandiges Bandpaß–Signal ergeben sich aber sehr viel geringere Anforderungen an den Hilbert–Transformator als bei einem breitbandigen Signal. Im Gegensatz zum Bild 1.1.1(b), muß hier lediglich sichergestellt werden, daß im Bereich Δf um f_0 die Hilbert–Transformationsbedingung gültig ist, siehe Bild 4.1.7.

$$H_{BP}(f) = -j\gamma(f) = \begin{cases} j & -f_0 - \frac{\Delta f}{2} < f < -f_0 + \frac{\Delta f}{2} \\ -j & f_0 - \frac{\Delta f}{2} < f < f_0 + \frac{\Delta f}{2} \end{cases} \tag{4.1.5}$$

Im folgenden werden Hilbert–Abtastempfänger verschiedener Näherung und Komplexität vorgestellt, deren Ordnung i sich aus der Anzahl der verwendeten Abtastwerte ergibt:

Ordnung $i \longrightarrow 2^i$–Abtastwerte

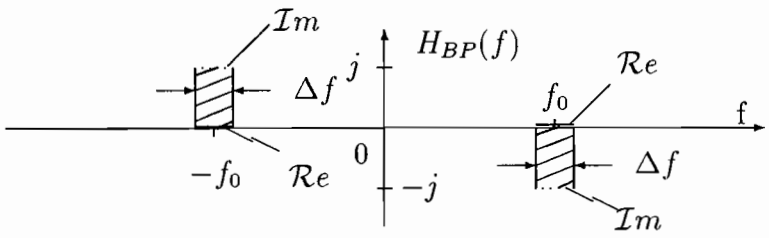


Bild 4.1.7 Anforderungen an den Hilbert–Transformator bei Bandpaß–Signalen.

4.1.4.1 Hilbert–Abtaster nullter Ordnung

Die Forderung aus Gleichung (4.1.5) wird schon recht gut durch eine analoge Laufzeitleitung mit der Laufzeit $T_L = \frac{1}{4f_0}$ angenähert †.

$$H_{\mathrm{BP}}(f) = e^{-j2\pi f T_L} = e^{-j\pi\frac{f}{2f_0}} = \cos\left(\frac{\pi f}{2f_0}\right) - j\sin\left(\frac{\pi f}{2f_0}\right) \tag{4.1.6}$$

Der Realteil bei diesem System wird bei $f = \pm f_0$ identisch Null und der Imaginärteil ergibt sich bei $f = \pm f_0$ zu $\mathcal{I}m\{H_{\mathrm{BP}}(\pm f_0)\} = \mp j$.

Dieser Ansatz läßt sich leicht in den Digitalbereich transformieren, siehe Bild 4.1.8.

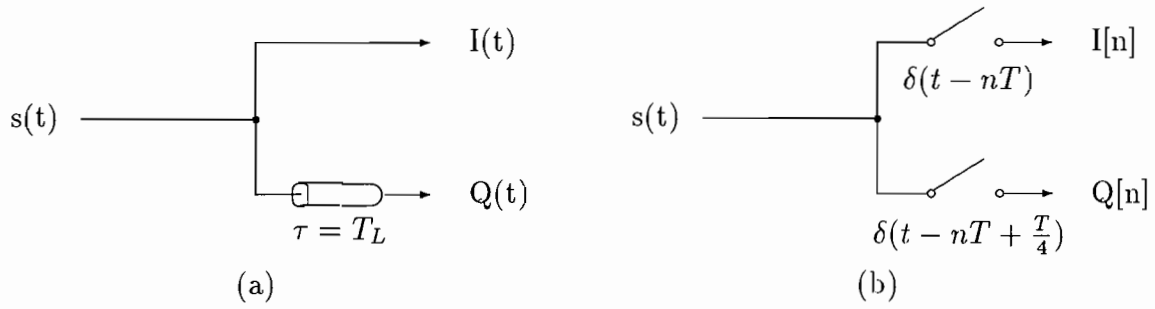


Bild 4.1.8 Analoge und digitale Realisierung des Hilbert–Transformators nullter Ordnung.

Der *Verzögerung* im Q–Zweig des analogen Hilbert–Transformators entspricht bei der Abtastung im Digitalbereich eine um T_L voreilende Abtastung.

Da hier jeweils nur ein Abtastwert zur Realisierung des Hilbert–Transformators benutzt wird, soll dieses System mit „Hilbert–Abtaster nullter Ordnung“ bezeichnet werden. Da der Hilbert–Abtaster nullter Ordnung und der Quadraturabtastempfänger nach [Wie88, S. 87-95] sehr ähnlich erscheinen, sei hier nochmals betont: Bei der Hilbert–Abtastung nullter Ordnung wird eine um 90° zur *Signalperiode* verschobene Abtastung vorgenommen, während bei der Quadratur–Abtastung eine zur *Abtastperiode* um 90° verschobene Abtastung vorgenommen wird.

† Dies läßt sich auch aus der zweiten Bezeichnung „90°–Phasenschieber“ für einen Hilbert–Transformator ableiten.

4.1.4.2 Hilbert–Abtaster höherer Ordnung

Der Hilbert–Abtaster nullter Ordnung liefert für sehr schmalbandige Signale recht gute Ergebnisse, die durch Hinzunehmen weiterer Abtastwerte (und eventuelle Gewichtung) noch verbessert werden können. Prinzipiell sind zwei verschiedene Strategien denkbar:

a) Nimmt man noch Terme mit größerer Laufzeit, die günstigerweise ungerade Vielfache der Verzögerung T_L sind, hinzu, siehe Bild 4.1.9(a), so wird eine Art Fourier–Reihenentwicklung der Hilbert–Spektralfunktion in dem betrachteten Bandpaß–Bereich durchgeführt. Diese Näherung soll als unsymmetrisch bezeichnet werden. Betrachtet man die I–Abtastung als Nullpunkt der Impulsgruppe, so liegt eine rein akausale Impulsgruppe vor.

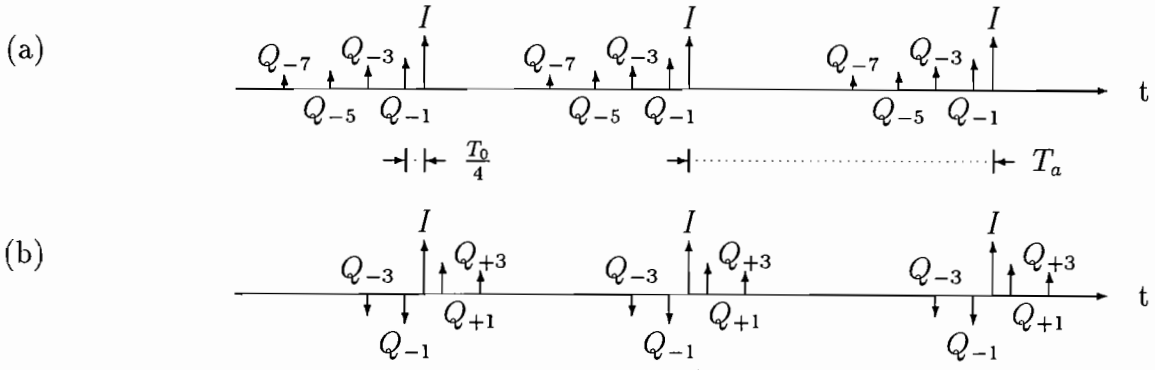


Bild 4.1.9 Abtastsequenz des Hilbert–Abtasters für den (a) unsymmetrischen und (b) symmetrischen Abtaster.

b) Man benutzt die Tatsache, daß eine ungerade reelle Zeitfunktion ein rein imaginäres ungerades Spektrum besitzt. Da der Realteil gleich Null ist, wird auch der entsprechende Fehler identisch Null. Die durch Bild 4.1.9(b) realisierte Abtastfolge soll als symmetrisch (sie ist genaugenommen schiefsymmetrisch) bezeichnet werden, da jede Impulsgruppe auch symmetrisch ist.

Als Verfahren zur Optimierung der Koeffizienten wurde ein modifiziertes Gradientenabstiegsverfahren benutzt, wie es z. B. in [EMR87, S. 170] beschrieben ist.

Nun seien noch die Ergebnisse der Optimierung kurz erläutert. Die Optimierung wurde für verschiedene Signalbandbreiten vorgenommen, wobei das Fehlermaß folgendermaßen gewählt wurde:

$$\text{Minimum des maximalen Fehlers} \qquad \{|H_{BP,n}(f) \mp j|\}$$
$$\pm \mathrm{f_0} - \tfrac{\Delta \mathrm{f}}{2} < \mathrm{f} < \pm \mathrm{f_0} + \tfrac{\Delta \mathrm{f}}{2}$$

Die Tabellen 4.1.3 bis 4.1.5 zeigen die Ergebnisse der Optimierung. Die verwendeten Modell–Signale waren:

a) DCF77–Signal mit $f_0 = 77,5$ kHz und $\Delta f = 20$ Hz

b) DCF77–Signal mit $f_0 = 77,5$ kHz und $\Delta f = 1$ kHz ‡

c) FAX–ZF–Signal mit $f_0 = 455$ kHz und $\Delta f = 6$ kHz

	Signal (a)	Signal (b)	Signal (c)
$i = 0$	$Q_{-1} = 1,0$	$Q_{-1} = 1,0$	$Q_{-1} = 1,0$
Max. Fehler in % :	$2,027 \cdot 10^{-2}$	$0,1097$	$0,8976$
$i = 1$	$Q_{-1} = 1,5029$	$Q_{-1} = 1,5023$	$Q_{-1} = 1,5023$
	$Q_{-3} = 0,5029$	$Q_{-3} = 0,5024$	$Q_{-3} = 0,5024$
Max. Fehler in % :	$1,2 \cdot 10^{-4}$	$8,49 \cdot 10^{-3}$	$8,83 \cdot 10^{-3}$
$i = 2$	$Q_{-1} = 1,5000097$	$Q_{-1} = 1,500046$	$Q_{-1} = 1,500048$
	$Q_{-3} = 0,5000099$	$Q_{-3} = 0,500015$	$Q_{-3} = 0,500016$
	$Q_{-5} = 0,0000002$	$Q_{-5} = -0,000108$	$Q_{-5} = -0,00011$
	$Q_{-7} = 0,0000096$	$Q_{-7} = -0,000076$	$Q_{-7} = -0,00008$
Max. Fehler in % :	$3,08 \cdot 10^{-6}$	$7,71 \cdot 10^{-3}$	$8,06 \cdot 10^{-3}$

Tabelle 4.1.3 Koeffizienten des unsymmetrischen Hilbert–Abtastempfängers.

Während in den Tabellen 4.1.3 und 4.1.4 die Ergebnisse einer rechnergenauen Optimierung dargestellt sind [MBH92], zeigt die Tabelle 4.1.5 das Resultat einer Vereinfachung für eine technisch günstige Realisierung[§]. Die Tabelle 4.1.5 enthält auch die maximale Bandbreite Δf in Hz für DCF77 Signale und das Verhältnis $\Delta f / f_0$ für beliebige Bandpaßsignale für verschiedene Anforderungen an die Auflösung in Bit. Ist eine n Bit Auflösung gefordert, so ergibt sich eine zu tolerierende Abweichung vom idealen Hilbert–Transformator von $E = 2^{-n} = 10^{-n*6/20}$.

[‡] Für die genaue Sekundensynchronisation sollte die Bandbreite entsprechend $B = \frac{1}{T} \approx \frac{1}{1ms} = 1$ kHz gewählt werden [Hil83, S. 49].

[§] Ein Koeffizient 0,5 läßt sich im binären Zahlensystem durch eine einfache Verschiebeoperation realisieren und ein Koeffizient 1,5 benötigt lediglich eine Verschiebeoperation und eine Addition.

	Signal (a)	Signal (b)	Signal (c)
$i = 1$	$Q_{-1} = 0,50000$	$Q_{-1} = 0,50001$	$Q_{-1} = 0,50001$
	$Q_{+1} = -0,50000$	$Q_{+1} = -0,50001$	$Q_{+1} = -0,50001$
Max. Fehler in % :	$1,03 \cdot 10^{-6}$	$2,57 \cdot 10^{-3}$	$2,68 \cdot 10^{-3}$
$i = 2$	$Q_{-1} = 0,56500$	$Q_{-1} = 0,56500$	$Q_{-1} = 0,56500$
	$Q_{+1} = -0,56500$	$Q_{+1} = -0,56500$	$Q_{+1} = -0,56500$
	$Q_{-3} = 0,06500$	$Q_{-3} = 0,06500$	$Q_{-3} = -0,06500$
	$Q_{+3} = -0,06500$	$Q_{+3} = -0,06500$	$Q_{+3} = -0,06500$
Max. Fehler in % :	$4,11 \cdot 10^{-8}$	$1,02 \cdot 10^{-4}$	$1,07 \cdot 10^{-4}$

Tabelle 4.1.4 Koeffizienten des symmetrischen Hilbert–Abtastempfängers.

Koeffizienten	Max. Fehler %	Auflösung/Bit	Δf-DCF77	$\Delta f/f_0$
$Q_{-1} = 1,0$	$3,98 \cdot 10^{-3}$	8	392,8	0,005069
$Q_{-1} = 1,0$	$2,51 \cdot 10^{-4}$	12	24,8	0,000320
$Q_{-1} = 1,0$	$1,58 \cdot 10^{-5}$	16	1,56	0,000020
$Q_{-1} = 1,5; Q_{-3} = 0,5$	$3,98 \cdot 10^{-3}$	8	2542,4	0,032805
$Q_{-1} = 1,5; Q_{-3} = 0,5$	$2,51 \cdot 10^{-4}$	12	638,5	0,008238
$Q_{-1} = 1,5; Q_{-3} = 0,5$	$1,58 \cdot 10^{-5}$	16	160,4	0,002069
$Q_1 = -0,5; Q_{-1} = 0,5$	$3,98 \cdot 10^{-3}$	8	4403,9	0,056825
$Q_1 = -0,5; Q_{-1} = 0,5$	$2,51 \cdot 10^{-4}$	12	1105,9	0,014269
$Q_1 = -0,5; Q_{-1} = 0,5$	$1,58 \cdot 10^{-5}$	16	277,8	0,003584

Tabelle 4.1.5 Koeffizienten einer technisch günstigen Realisierung des Hilbert–Abtastempfängers.

Es zeigt sich im oberen Teil von Tabelle 4.1.4, daß bereits der symmetrische Hilbert–Abtaster erster Ordnung, (d. h. ein I–Abtastwert mit Gewicht 1 und zwei Q–Abtastwerte mit einem Gewicht von $\pm 0,5$), bei allen Beispielsignalen für eine 12 Bit Quantisierung vollkommen ausreichend ist.

Für die technisch günstigen Koeffizienten von Tabelle 4.1.5 sind in den Bildern 4.1.10 bis 4.1.12 die zugehörigen Spektralfunktionen und die Abweichung vom idealen Hilbert–Transformator für Real– und Imaginärteil dargestellt. Ebenfall sind in den drei

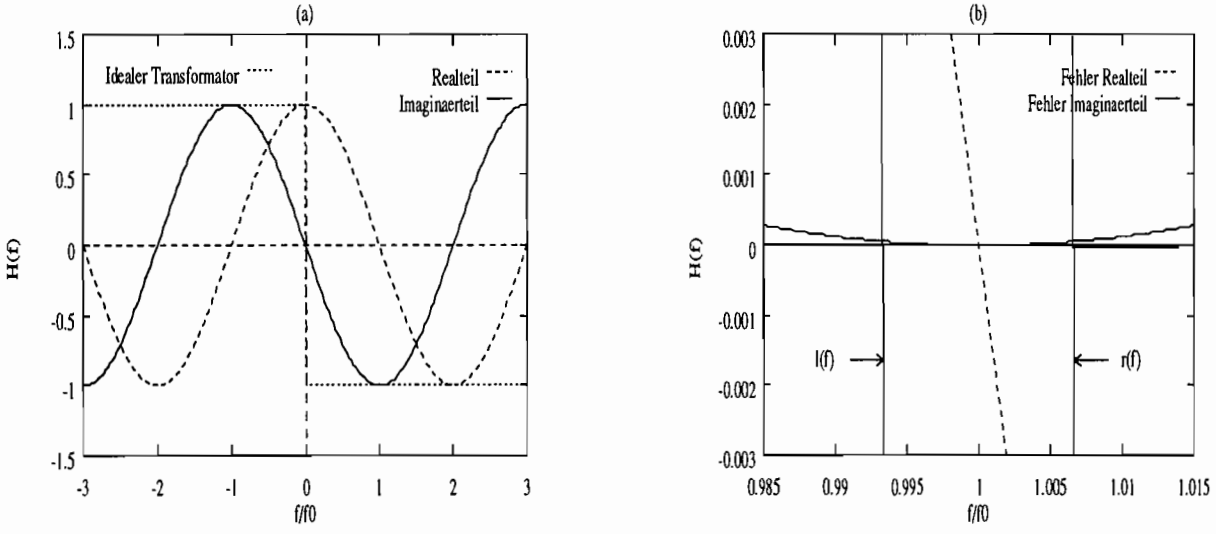


Bild 4.1.10 Hilbert–Transformator nullter Ordnung mit Koeffizient $Q_{-1} = 1,0$. (a) Spektrum. (b) Abweichung vom idealen Transformator.

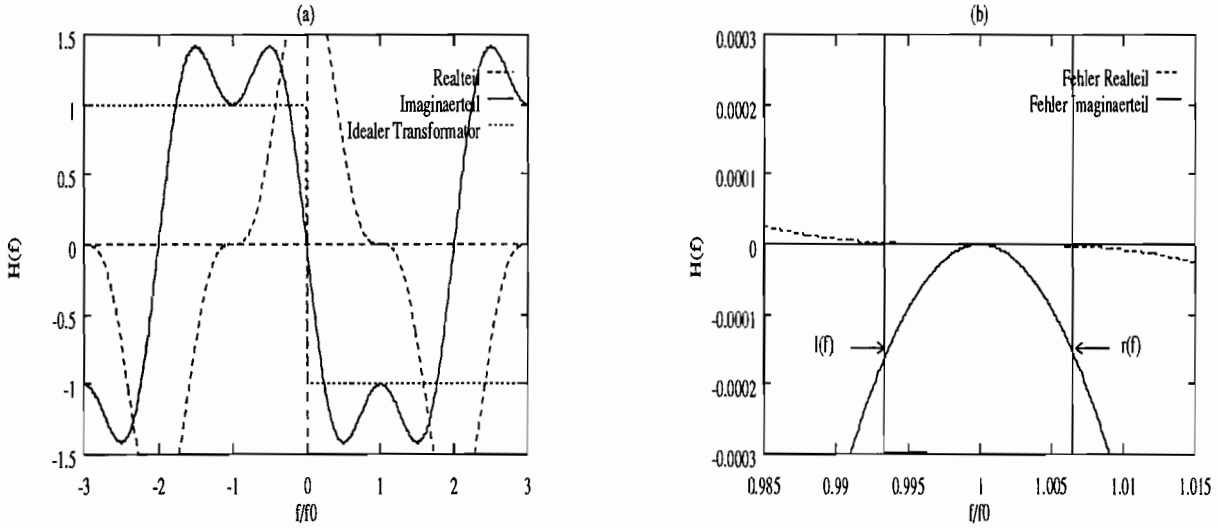


Bild 4.1.11 Unsymmetrischer Hilbert–Transformator erster Ordnung mit Koeffizienten $Q_{-1} = 1,5$; $Q_{-3} = 0,5$. (a) Spektrum. (b) Abweichung vom idealen Transformator.

Teilbildern (b) jeweils $l(f)$ und $r(f)$ als linke und rechte Schranke der Bandbreite von Modellsignal **c)** mit eingezeichnet.

Weiterhin ist interessant, daß die Optimierung der Koeffizienten, die mit $\frac{1}{1000}$ Genauigkeit dargestellt sind, für alle drei Signaltypen **(a)** bis **(c)** identische Ergebnisse lieferten, obwohl die Verhältnisse $\Delta f/f_0$ um bis zu einem Faktor 50 differieren!

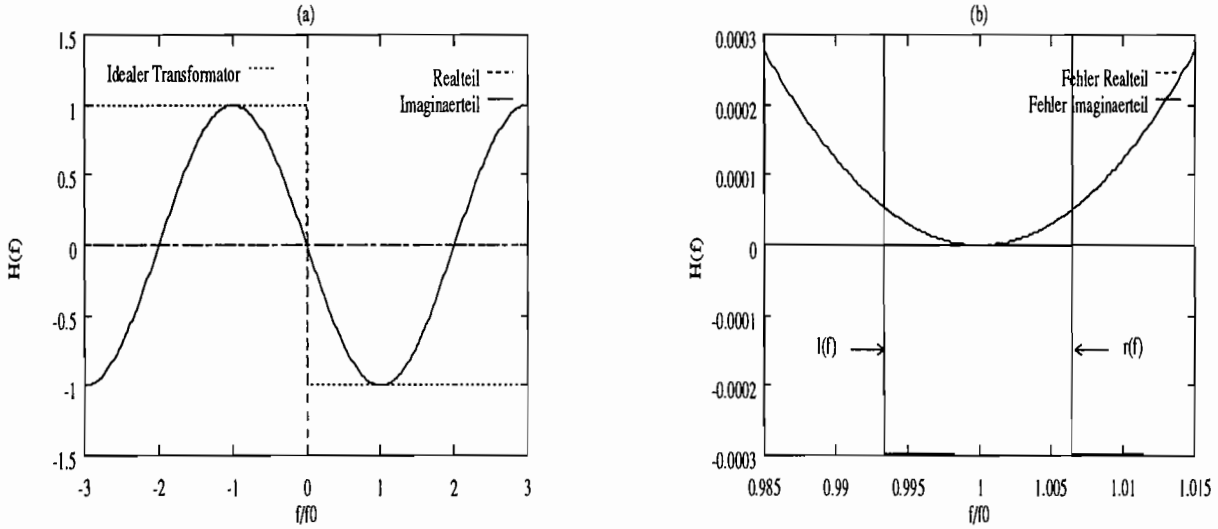


Bild 4.1.12 Symmetrischer Hilbert–Transformator erster Ordnung mit Koeffizienten $Q_1 = -0,5$; $Q_{-1} = 0,5$. (a) Spektrum. (b) Abweichung vom idealen Transformator.

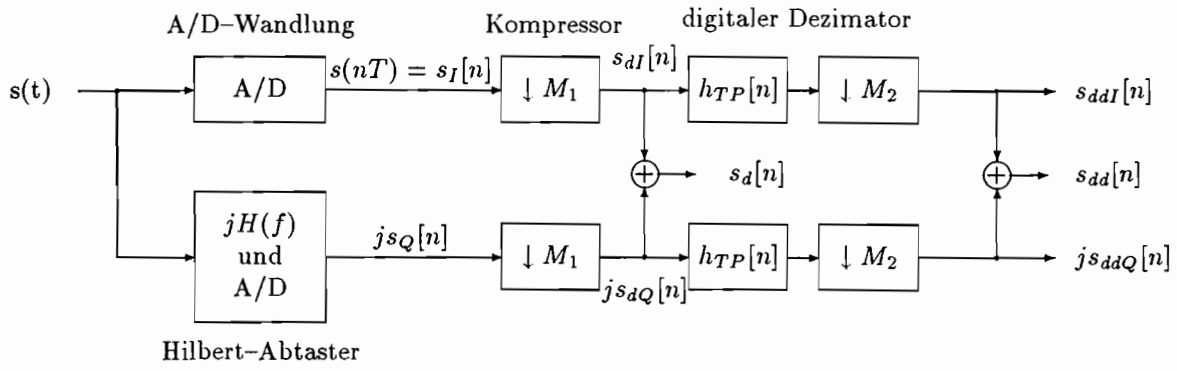


Bild 4.1.13 Hilbert–Abtastempfänger mit Unterabtastung.

4.1.4.3 Hilbert–Abtastempfänger mit Unterabtastung

Wie im letzten Abschnitt erläutert, läßt sich ein Hilbert–Abtastempfänger für Bandpaß–Signale mit relativ geringem Fehler und wenig Aufwand realisieren. Setzen wir nun den

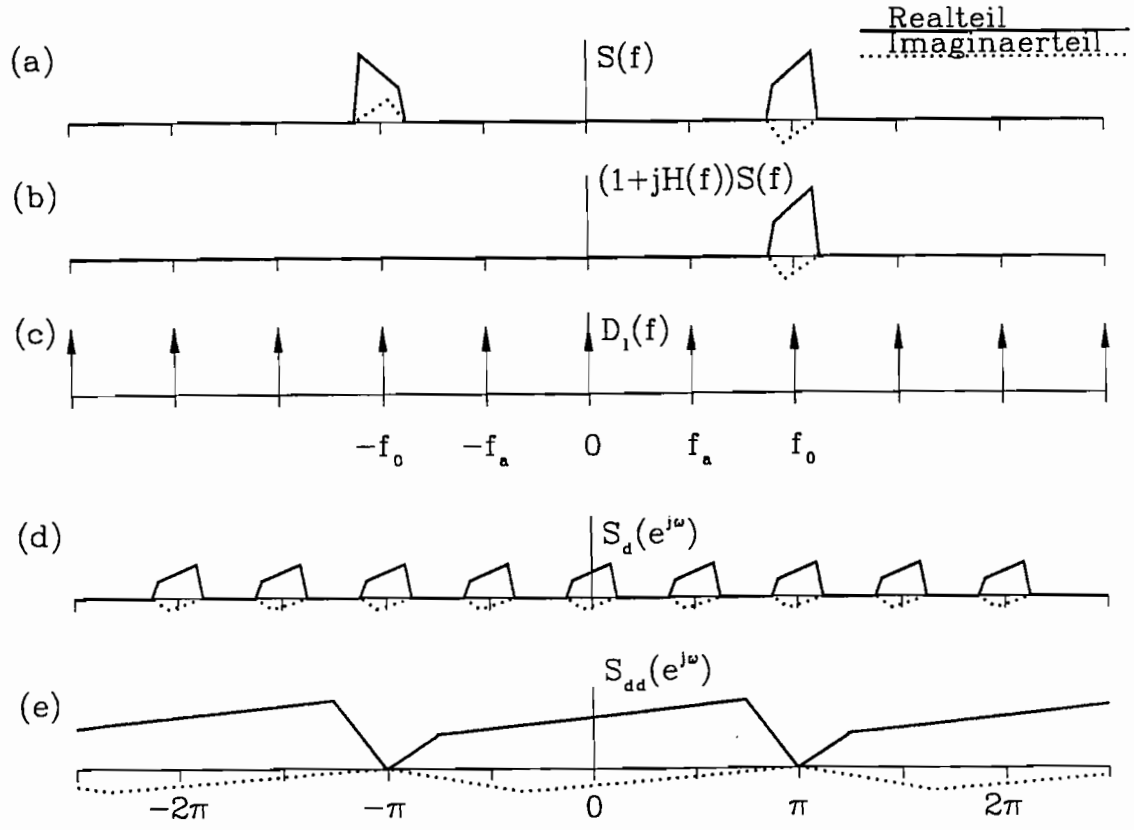


Bild 4.1.14 Spektrale Betrachtung zum Hilbert–Abtastempfänger mit Unterabtastung um den Faktor $M = 2$.

Hilbert–Abtaster wie in Bild 4.1.13 ein, so ergibt sich ein komplexes analytisches Signal. Dieses Signal kann nun reduziert werden, ohne daß eine „Integer–Band"–Verletzung auftreten kann, da ja ein analytisches Signal vorliegt, siehe Bild 4.1.14. Da die Hilbert–Transformation mit einem reinen Abtastempfänger (mit gewichteten Koeffizienten) durchgeführt wird, kann die Abtastratenreduktion $\downarrow M_1$ unmittelbar mit in die Abtastung hineingenommen werden. Im einzelnen zeigt Bild 4.1.14(a) das (bandpaßgefilterte) Signal am Eingang des Empfängers. In Bild 4.1.14(b) ist die Summe aus der Hilbert–Transformierten und dem Eingangssignal selbst dargestellt, wie sie aufgrund der speziellen Lage der Abtastzeitpunkte des Hilbert–Abtasters entsteht. Da es durch A/D–Wandlung und Abtastratenreduktion um $\downarrow M_1$ durch die Unterabtastung zu einer periodischen Fortsetzung des analytischen Spektrums aus Bild 4.1.14(b) kommt, ist im Spektralbereich eine Faltung mit $D_{\mathrm{I}}(f)$ auszuführen, was Bild 4.1.14(d) zeigt. Das nun vorliegende digitale Signal $s_{\mathrm{dI}}[n] + j s_{\mathrm{dQ}}[n] = s_{\mathrm{d}}[n] \circ\!\!-\!\!\bullet\ S_{\mathrm{d}}(e^{j\omega})$ *kann* nun (nach einer eventuell nötigen digitalen Filterung) im einer zweiten Dezimationsstufe um $\downarrow M_2$ weiter dezimiert[¶] werden, bis die Wiederholspektren aneinandergrenzen, siehe Bild 4.1.14(e). Insgesamt wurde somit eine Reduktion um $\downarrow M = M_1 \cdot M_2$ vorgenommen. Am Ausgang des Empfängers entsteht nun das digital dezimierte Signal $s_{\mathrm{dd}}[n]$ mit zugehörigem Spektrum $S_{\mathrm{dd}}(e^{j\omega})$.

Die Bilder 4.1.15(a) bzw. (b) zeigen zwei Alternativen des Hilbert–Abtasters erster Ordnung, die in den Arbeiten [Sch92] bzw. [Die92] realisiert wurden. Der Unterabtastfaktor kann bei den Empfängern (über einen DIP–Schalter) im Bereich $\downarrow M = 2 \ldots 256$ variiert werden. Die Vorselektion wird im Antennenkreis durchgeführt. Hauptselektion leistet ein DSP, siehe Tabelle 4.5.2.

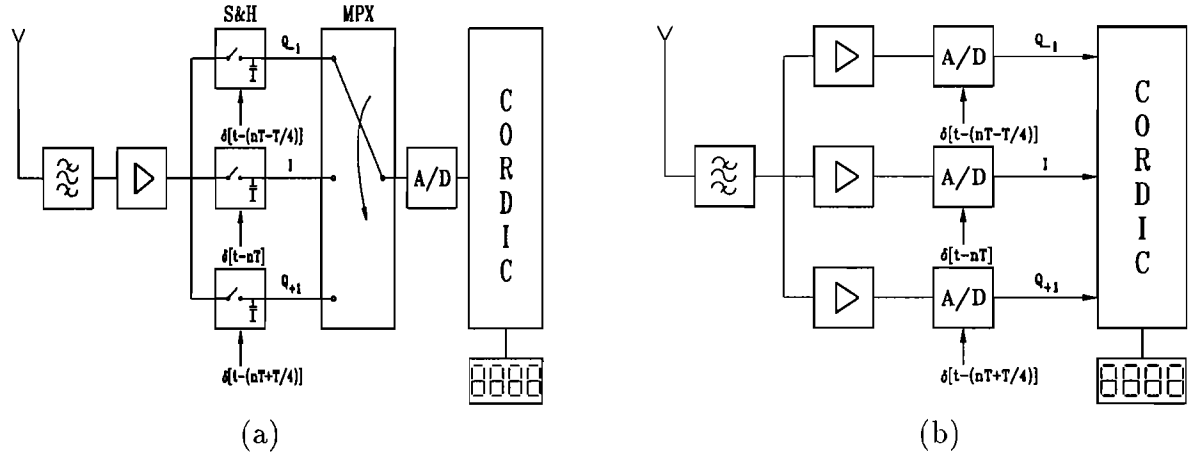


Bild 4.1.15 (a) Hilbert–Abtastempfänger erster Ordnung mit einer mehrfach S&H–Stufe und einem A/D–Wandler im Multiplex [Sch92] und (b) Alternative mit 3 A/D–Wandlern bei gleichzeitiger starker Unterabtastung.

Im Gegensatz zur Hilbert–Abtastung haben die Untersuchungen des Quadraturabtastempfängers bei Unterabtastung [MB91, S. 15-20] gezeigt, daß es in der ersten Reduk-

[¶] Es kann natürlich auch bereits durch den Hilbert–Abtaster eine größtmögliche Reduktion um $M = M_1$, d. h. $M_2 = 1$ realisiert sein. Dies erhöht allerdings die Anforderung an die analoge Filterung des Eingangssignals eines Empfängers.

tionsstufe M_1 sehr wohl zu einer „Integer–Band"–Verletzung und damit zu nicht wieder rekonstruierbaren Verzerrungen des Signals kommen kann, siehe Bild 4.1.6.

4.1.5 Vergleich der Abtastempfänger

Der Vergleich der Abtastempfänger läßt sich folgendermaßen zusammenfassen:

- Durch die Erzeugung einer I– und Q–Phase erhält man ein komplexes analytisches Signal.
- Abtastempfänger sind geeigneter als der (analoge) Quadraturüberlagerungsempfänger zur Erzeugung der I– und Q–Phase, da die Multiplikation durch einfache Abtastung ersetzt wird.
- „Integer–Band"–Abtastung und Quadraturabtastung erfordern eine sorgfältige Wahl der Parameter M, k und f_A bei Unterabtastung, damit es zu keiner „Integer–Band"–Verletzung kommt.
- Der Hilbert–Transformator läßt sich sehr gut durch einen Hilbert–Abtaster annähern.
- Der Hilbert–Abtastempfänger eignet sich besser als die Quadraturabtastung bei Unterabtastung, da keine weitere Filterung zur Erzeugung von I– und Q–Phase nötig ist und es außerdem bei der Hilbert–Abtastung zu keiner „Integer–Band"–Verletzung kommen kann.

4.2 Demodulation von Bandpaßsignalen mit Hilfe des CORDIC–Algorithmus

In den Abschnitten 4.1.1 bis 4.1.4 wurde ausführlich dargestellt, wie aus einem Bandpaß–Signal durch Mischung oder Abtastung eine I– und Q–Phase erzeugt wird, deren komplexe Summe $I \pm jQ$ ein analytisches Signal bildet. Ziel und Zweck einer Demodulation ist es, mit Hilfe dieser I– und Q–Phase bei den jeweiligen Modulationsarten das Sendesignal zurückzugewinnen. Hier bietet der CORDIC–Algorithmus, wie nachfolgend gezeigt wird, eine ideale Basis für die Demodulation sowohl von AM–, PM– als auch FM–Signalen.

4.2.1 Grundlagen des COordinate Rotation DIgital Computer (CORDIC)–Prinzips

Das COordinate Rotation DIgital Computer (CORDIC)–Prinzip wurde erstmals von J. E. Volder [Vol59] im September 1959 vorgeschlagen und ist ein sehr universeller Algorithmus zur Berechnung von trigonometrischen und hyperbolischen Funktionen und deren Umkehrfunktionen [Sch74, S. 162-176, S. 181-193], [Wal71] (z. B.: sin, cos, tan,

arctan, arcsin, arccos, sinh, cosh). Der Algorithmus ist sowohl im Binär– als auch im Dezimalzahlensystem einsetzbar. †

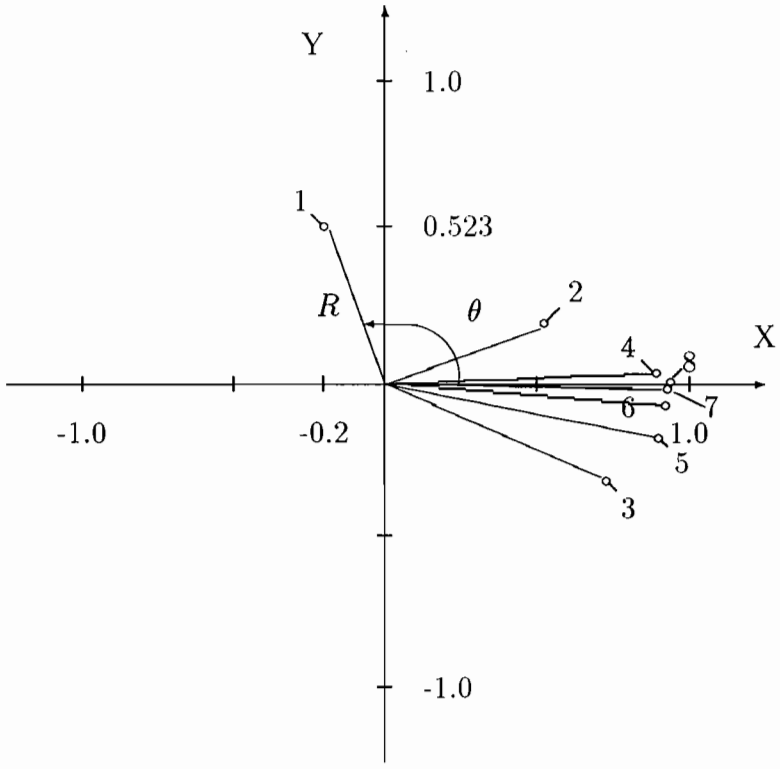
Von besonderem Interesse für den Einsatz als Demodulator ist die Binärzahlen–Darstellung, die das direkte Ergebnis der A/D–Wandlung ist. Weiterhin wird von der Möglichkeit, das CORDIC–Prinzip zur Transformation von rechtwinkligen in polare Koordinaten $(X, Y \rightarrow R, \theta)$ einzusetzen, Gebrauch gemacht, da dies unmittelbar eine AM– bzw. PM–Demodulation darstellt. Eine klassische Berechnung der polaren Koordinaten mit $R = \sqrt{X^2 + Y^2}$ und $\theta = \arctan(Y/X)$ bedarf einer umfangreichen Berechnung und wird durch den CORDIC–Algorithmus wesentlich vereinfacht [Hah91, S. 81-83].

4.2.2 Funktionale Beschreibung des Koordinatentransformators

Der CORDIC–Algorithmus ist eine iterative Prozedur, bei der ein Zeiger in der X, Y–Ebene in jedem Schritt um einen bestimmten Winkel $\pm\alpha_i$ rotiert wird. Hierbei

† Es wird von „Insidern“ behauptet, daß der dezimale CORDIC–Algorithmus in den wissenschaftlichen Taschenrechnern von HP verwendet wird [Sch74].

$$
\begin{aligned}
\alpha_1 &= \arctan(\infty) = 90^\circ \\
\alpha_2 &= \arctan(2^0) = 45^\circ \\
\alpha_3 &= \arctan(2^{-1}) = 26,5^\circ \\
\alpha_4 &= \arctan(2^{-2}) = 14,0^\circ \\
&\vdots \\
\alpha_i &= \arctan(2^{-(i-2)}) \quad \forall i > 1
\end{aligned}
$$



(a) (b)

Bild 4.2.1 (a) Zur Definition der Winkel α_i. (b) Beispiel für die Vektorisierung

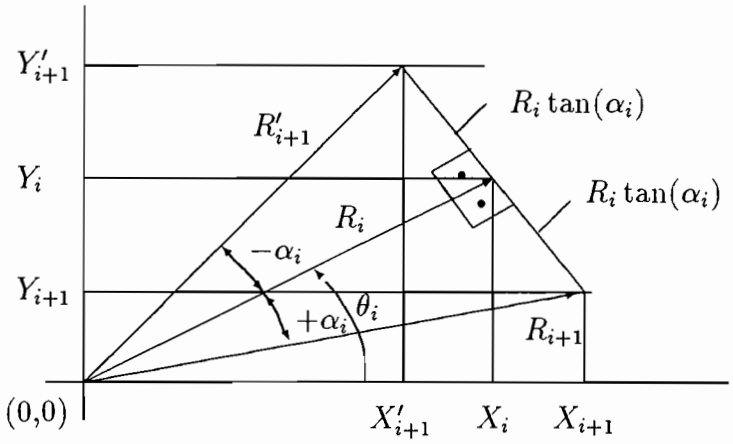


Bild 4.2.2 Rotationsprinzip des CORDIC–Algorithmus.

unterscheidet man die **Vektorisierung** $(X, Y \rightarrow R, \theta)$, die der Demodulation entspricht, und die **Rotation** $(R, \theta \rightarrow X, Y)$, die einer Modulation entspricht.

Im folgenden soll nur die Vektorisierung betrachtet werden, da eine Berechnung von (R, θ) zu realisieren ist. Anhand des Beispiels aus Bild 4.2.1 sei das prinzipielle Vorgehen erläutert. Ausgangspunkt ist der Vektor „1“, der als X– und Y–Koordinatenwert gegeben ist. Dieser Vektor wird nun in jedem Iterationsschritt um einen Winkel $\pm\alpha_i$ so gedreht, daß er letztendlich auf der X–Achse zu liegen kommt. Der akkumulierte Winkel $\sum_{i=1}^{i=n} \alpha_i$ stellt die gesuchte Phase dar, während der Wert von X_n dem gesuchten Radius entspricht.

Die ausschlaggebende Idee von Volder [Vol59] für die Funktion und das Verständnis des Algorithmus liegt in der Wahl der Folge von Drehwinkeln (entweder im positiven oder negativen Drehsinn), siehe Bild 4.2.1(a).

Da die arctan-Funktion im Bereich $0 \cdots 45°$ immer oberhalb der Winkelhalbierenden liegt, folgt, daß immer $\alpha_{i+1} \geq \frac{\alpha_i}{2}$ erfüllt ist. Die Drehung um 90° im ersten Iterationsschritt unter Berücksichtigung des Vorzeichens von Y bewirkt dabei, daß alle Vektoren im 1. oder 4. Quadranten zu liegen kommen. Da die Summe $\sum |\alpha_i|$ gegen 180° strebt, ist gewährleistet, daß alle Winkel im Bereich $-180° \cdots 180°$ erreicht werden können.

Anhand des Bildes 4.2.2 soll nun die Wirkungsweise der Rotation um einen Winkel $\pm\alpha_i$ erklärt werden:

Der Vektor R_i, θ_i mit den Koordinaten X_i, Y_i werde um den Winkel $\alpha_i = \arctan(2^{-(i-2)})$ auf die neuen Koordinaten X_{i+1}, Y_{i+1} gedreht.

Mit Hilfe der ersten Iteration‡ (Drehung um 90°) wird erreicht, daß der Zeiger X_2, Y_2 im 1. oder 4. Quadranten liegt. Bei $X_1 \geq 0$ müssen daher keine Operationen ausgeführt werden. Bei $X_1 < 0$ müssen Vertauschungen von X– und Y–Registerinhalten ($X_2 = \pm Y_1$; $Y_2 = \mp X_1$; $\theta_2 = \pm\alpha_1$) in Abhängigkeit des Vorzeichen des Y–Registers vorgenommen werden, vergleiche Bild 4.2.1(b).

‡ Da die erste Iteration ein Ausnahme darstellt, kann es bei speziellen Hardware–Realisierungen (Zustandsmaschine, siehe die Bilder 4.2.4 und 4.2.5) günstiger sein, vollständig auf die erste Iterationsschritt zu verzichten und nach der letzten Iteration ein Korrekturschritt durchzuführen, siehe hierzu [Hil92b, S. 33-34] oder [Tim90, S. 49].

Für die Iterationen $i > 1$ ist die Drehung so definiert, daß das Dreieck $(0,0)(X_i, Y_i)$ (X_{i+1}, Y_{i+1}), das zu $+\alpha_i$ (Rechtsdrehung) gehört, und das Dreieck $(0,0)$ (X_i, Y_i) (X'_{i+1}, Y'_{i+1}), das zu $-\alpha_i$ (Linksdrehung) gehört, *rechtwinklig* sind. Für den neuen Radius R_{i+1} ergibt sich:

$$R_{i+1}^2 = R_i^2 + \tan^2(\alpha_i)\, R_i^2 \tag{4.2.1}$$

Mit der Wahl von α_i zu $\alpha_i = \arctan(2^{-(i-2)})$ folgt:

$$R_{i+1} = \sqrt{1 + 2^{-2(i-2)}}\, R_i \tag{4.2.2}$$

Zu erkennen ist, daß der Radius des Vektors R_{i+1} um den Wurzel–Faktor größer als R_i ist, was einer linearen Dehnung entspricht. Die neuen X, Y–Koordinaten ergeben sich wie folgt:

$$X_{i+1} = R_{i+1}\cos(\theta_i \pm \alpha_i) \tag{4.2.3}$$

$$= R_i\sqrt{1 + 2^{-2(i-2)}}\, [\cos(\theta_i)\cos(\alpha_i) \mp \sin(\theta_i)\sin(\alpha_i)] \tag{4.2.4}$$

Mit den trigonometrischen Beziehungen [BS84, S. 185]

$$\cos[\arctan(\psi)] = \frac{1}{\sqrt{1+\psi^2}} \tag{4.2.5}$$

$$\sin[\arctan(\psi)] = \frac{\psi}{\sqrt{1+\psi^2}} \tag{4.2.6}$$

folgt beim Einsetzen in Gleichung (4.2.4)

$$X_{i+1} = R_i\sqrt{1 + 2^{-2(i-2)}} \left[\cos(\theta_i)\frac{1}{\sqrt{1 + 2^{-2(i-2)}}} \mp \sin(\theta_i)\sqrt{\frac{2^{-2(i-2)}}{1 + 2^{-2(i-2)}}}\right]$$

$$= \underbrace{R_i\cos(\theta_i)}_{X_i} \mp \underbrace{R_i\sin(\theta_i)}_{Y_i} 2^{-(i-2)}$$

$$X_{i+1} = X_i \mp Y_i 2^{-(i-2)} \tag{4.2.7}$$

Die Berechnung für Y_{i+1} erfolgt in analoger Weise:

$$Y_{i+1} = R_{i+1}\sin(\theta_i \pm \alpha_i)$$

$$Y_{i+1} = Y_i \pm X_i 2^{-(i-2)} \tag{4.2.8}$$

Mit Hilfe der Gleichungen (4.2.7) und (4.2.8) kann man fortgesetzt den Vektor (R_i, θ_i) so um $\pm\alpha_i$ drehen, bis $Y_n = 0$ wird. Welches Vorzeichen man nehmen muß, entscheidet sich durch das Vorzeichen von Y in folgender Weise:

Ist das Vorzeichen des Registers Y positiv (negativ), so muß im Uhrzeigersinn (gegen den Uhrzeiger), d. h. mathematisch negativ (positiv), gedreht werden, und in den

Gleichungen (4.2.7) und (4.2.8) müssen die unteren (oberen) Vorzeichen benutzt werden.

Der akkumulierte Winkel $\sum_{i=1}^{i=n} \alpha_i$ stellt die gesuchte Phase θ dar, während der Wert von X_n dem gesuchten Radius R entspricht. Dieser Radius ist allerdings um einen linearen Faktor K verzerrt [§] , was aber bei Verwendung des CORDIC–Algorithmus als Demodulator nur insoweit berücksichtigt werden muß, als das X–Register eventuell um ein Bit größer gewählt werden muß, als es dem maximalen Eingangswert des X–Registers entspricht, damit es während der Iteration zu keinem Überlauf kommt.

Zusammenfassend ergeben sich die Iterationsgleichungen zu:

$$X_{i+1} = X_i \mp Y_i 2^{-(i-2)} \tag{4.2.11}$$

$$Y_{i+1} = Y_i \pm X_i 2^{-(i-2)} \tag{4.2.12}$$

$$\theta_{i+1} = \theta_i \mp \alpha_i \tag{4.2.13}$$

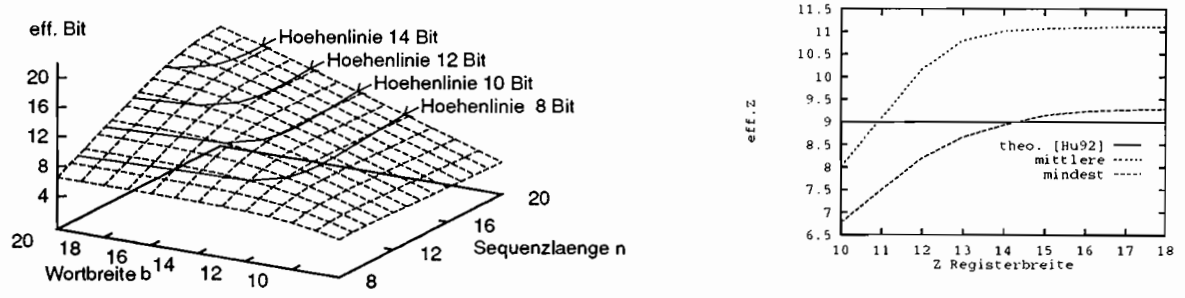
Man erkennt, daß die Berechnung von X_{i+1} und Y_{i+1} schaltungstechnisch sehr einfach (Verschiebe–Operation und Addition) durchzuführen sind.

[§] Bei Verwendung in einem Taschenrechner muß der Korrekturfaktor

$$K = \prod_{i=0}^{n} \sqrt{1 + \tan^2(\alpha_i)} \tag{4.2.9}$$

$$= \prod_{i=0}^{n} \frac{1}{\cos(\alpha_i)} \approx 1,6468 \quad \forall \quad n \geq 8 \tag{4.2.10}$$

durch Multiplikation mit $1/K$ oder Division durch K berücksichtigt werden.



(a) (b)

Bild 4.2.3 (a) Anzahl der effektiver Bit entsprechend der theoretischen Schranke entsprechend Gleichung (4.2.14). (b) Einfluß der Z–Register–Breite auf die effektive Z–Auflösung bei 13 Bit X/Y–Auslösung und 8 Iterationen.

	Aussteuerung	Minimum	Durchschnitt	Varianz	nach [Hu92]
Betrag	25% – 100%	10,160	11,160	0,470	9,648
Winkel	25% – 100%	9,143	11,071	1,306	9,000
Betrag	50% – 100%	10,160	11,121	0,453	9,648
Winkel	50% – 100%	9,277	11,080	1,309	9,000
Betrag	100%	10,161	11,061	0,414	9,648
Winkel	100%	9,538	10,990	1,286	9,000

Tabelle 4.2.1 Effektive Bit–Breiten des CORDIC–Prozessors bei 13 Bit–Breite X/Y–Pfad, 15 Bit–Breite Z–Pfad und 8 Iterationen.

Frei ist die Wahl der Bit–Breiten und der Iterationstiefe. Diese Größen sind vom zu akzeptierenden Hardware–Aufwand und der geforderten Genauigkeit abhängig. Für die effektiven Bit in R liefert die theoretische Abschätzung nach [Hu92] brauchbare Ergebnisse

$$d_{\text{eff}} = log_2 \left(\frac{1}{2^{-n+n'} |\vec{v}(0)| + 2^{-b-0.5} \left(G_{\text{m}}(n)/K_{\text{m}} + 1 \right)} \right) - 1 \qquad (4.2.14)$$

Diese Gleichung ist in Bild 4.2.3 für Sequenzlängen n im Bereich 8 bis 20 und Bit–Breiten im X/Y–Pfad b von 8-20 bei Vollaussteuerung $|\vec{v}(0)|_{\text{max}}$ veranschaulicht.

Die Untersuchungen in [Hil92b] haben jedoch gezeigt, daß im Gegensatz zu bisherigen Veröffentlichungen [Hu92] die Winkelauflösung (Z bzw. θ) nicht ausschließlich von der Anzahl der Iterationen, Betrag des Eingangswertes und Auflösung im X/Y–Pfad abhängt, sondern zusätzlich die Rundungsfehler im Winkelregister selbst und Rundungsfehler in Z aufgrund der Rundungsfehler im X/Y–Pfad berücksichtigt werden müssen. Insbesondere konnten die Simulationen den Nachweis erbringen, daß die Winkelauflösung nur asymptotisch die Betragsauflösung erreicht, siehe Bild 4.2.3(b).

Abschließend sei darauf hingewiesen, daß die tatsächlich realisierte effektive Bit–Breite bei Aussteuerung 25% – 100%, wie sie z. B. beim AM–Signalanteil von DCF77 auftritt, nicht wesentlich schlechter ist als bei der reinen PM/FM (100% Aussteuerung). Die Realisierung in [Hil92b] besitzt effektive und mittlere Bit–Breiten entsprechend Tabelle 4.2.1.

4.2.3 Mögliche Hardware–Realisierung des CORDIC–Algorithmus

Der CORDIC–Algorithmus läßt sich sowohl in einer Zustandsmaschine als auch in einer vollständigen Pipeline–Struktur realisieren [Hil92b], [Nol91].

Beide Architekturen lassen sich vorteilhaft mit einem programmierbaren Gate Array[¶] realisieren. Kommt es auf maximale Verarbeitungsgeschwindigkeit an, so wird jede Berechnungsstufe des Algorithmus in Hardware realisiert, was bei einer Bit–Breite b bereits

[¶] Die angegeben CLB–Zahlen (Configurable Logic Block) beziehen sich auf die Serie XC3000 der Firma Xilinx.

Struktur	Register	Multiplexer	Addierer	Shifter	$\sum$CLB	Takte
2BS+2Ak	2*7	0	2*14	2*19,5	81	12
2RS+2Ak	2*7	0	2*14	2*6,5	55	46
2LRS+2Ak	2*7	0	2*14	2*8	58	39
1BS+2Ak	7	3*7	2*14	19,5	75,5	20
1RS+2Ak	7	3*7	2*14	6,5	62,5	56
1LRS+2Ak	7	3*7	2*14	8	64	74
1BS+1Ak	3*7	2*7	14	19,5	68,5	20
1RS+1Ak	3*7	2*7	14	6,5	55,5	92
1LRS+1Ak	3*7	2*7	14	8	57	74

Tabelle 4.2.2 CLB–Aufwandsabschätzung für 13 Bit plus Vorzeichen–Bit X/Y–Pfad. (Abkürzungen:Ak=Akkumulator; BS=Barrelshifter; RS=serieller Rechts–Shifter; LRS=serieller Links/Rechts–Shifter)

auf ($[b-1]$–Stufen der Pipeline $\times 3 \times b$)–CLBs führt. Bei $b = 8$ Bit werden so 168 CLBs benötigt, siehe Bild 4.2.7.

In Bild 4.2.4 wird eine Zustandsmaschine mit Register–Rechenwerk vorgestellt, die dann bevorzugt wird, wenn möglichst wenig Platz in dem PGA–Baustein verbraucht werden soll. Für eine weitere Aufwandsverringerung kann die „vollparallele Struktur" [Tim90, S. 63] aus Bild 4.2.4 in eine „schlanke Struktur" überführt werden (nur ein Shifter und/oder nur ein Akkumulator, siehe Bild 4.2.5). Weiterhin besteht die Möglichkeit, den aufwendigen Barrelshifter (BS) durch einen seriellen Rechts–Shifter (RS) bzw. einen

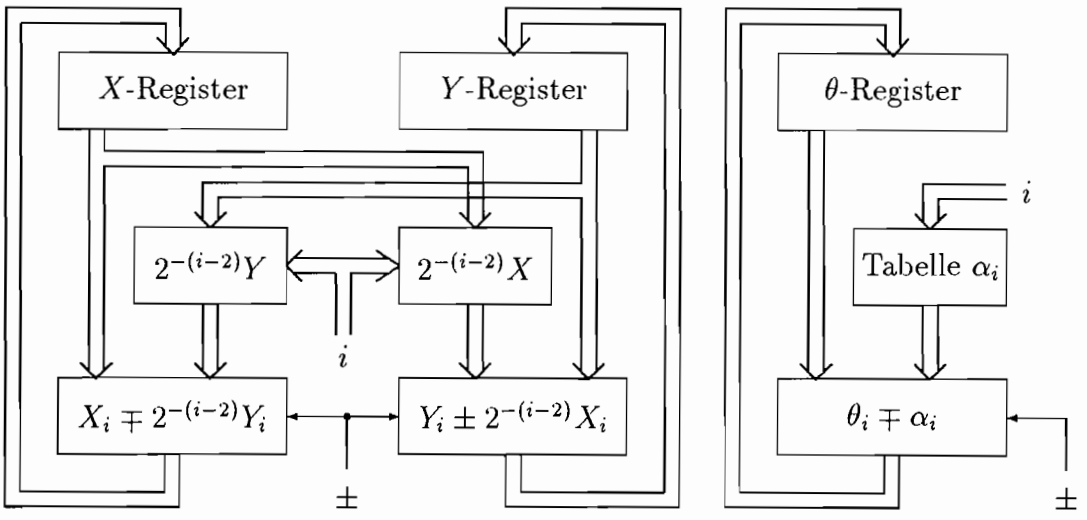


Bild 4.2.4 Realisierung des CORDIC–Algorithmus als „vollparallele" Zustandsmaschine.

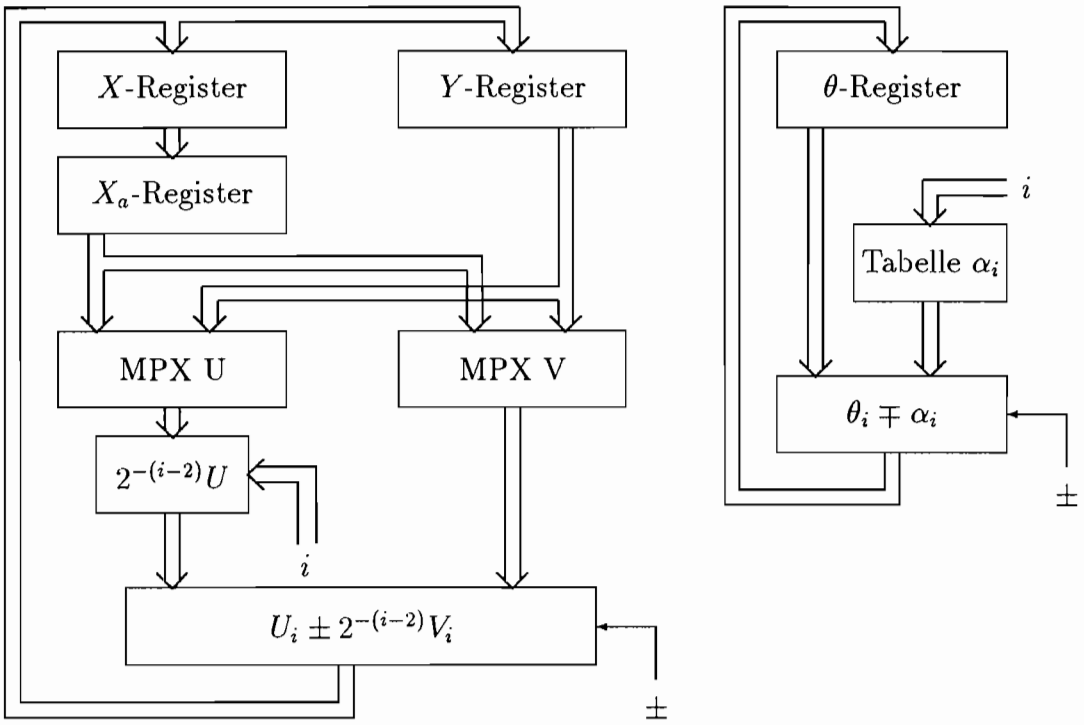


Bild 4.2.5 Realisierung des CORDIC–Algorithmus in der „schlanken Struktur".

seriellen Links/Rechts–Shifter (LRS) zu ersetzen. Für die verschiedenen Architekturen

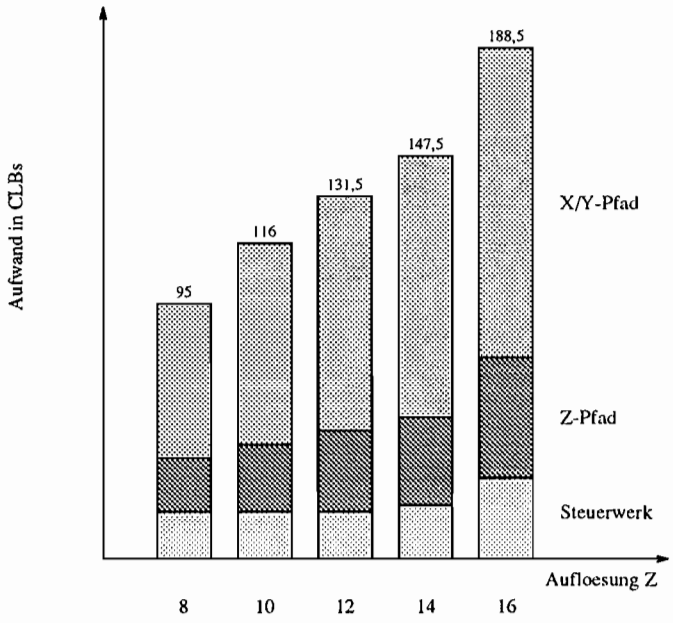


Bild 4.2.6 Aufwand bei verschiedener Anforderung an die Auflösung im Z–Register bei Realisierung der „schlanke Struktur" mit einem Barrelshifter und einem Akkumulator.

kann man den Aufwand an CLBs für den X/Y-Pfad und die resultierende Latenzzeit bei Realisierung mit einem Xilinx PGA der Tabelle 4.2.2 entnehmen. Als ein guter Kompromiß zwischen kleiner Latenzzeit und geringem Aufwand erscheint hierbei die „schlanke Struktur“ mit einem Barrelshifter und einem Akkumulator.

In Bild 4.2.6 ist der Aufwand veranschaulicht, der für die Realisierung in der „schlanken Struktur“ (1BS+1Ak) bei geforderten effektiven 8-16 Bit nötig ist. Deutlich wird, daß mit linear steigender Anforderung an die Z-Auflösung lediglich ein linear anwachsender Aufwand an CLBs entsteht. Der Sprung im Aufwand zwischen 14 und 16 Bit Auflösung resultiert aus den auf 5 Eingangsvariablen beschränkten logischen Verknüpfungen in einem „Configurable Logic Block“ (CLB).

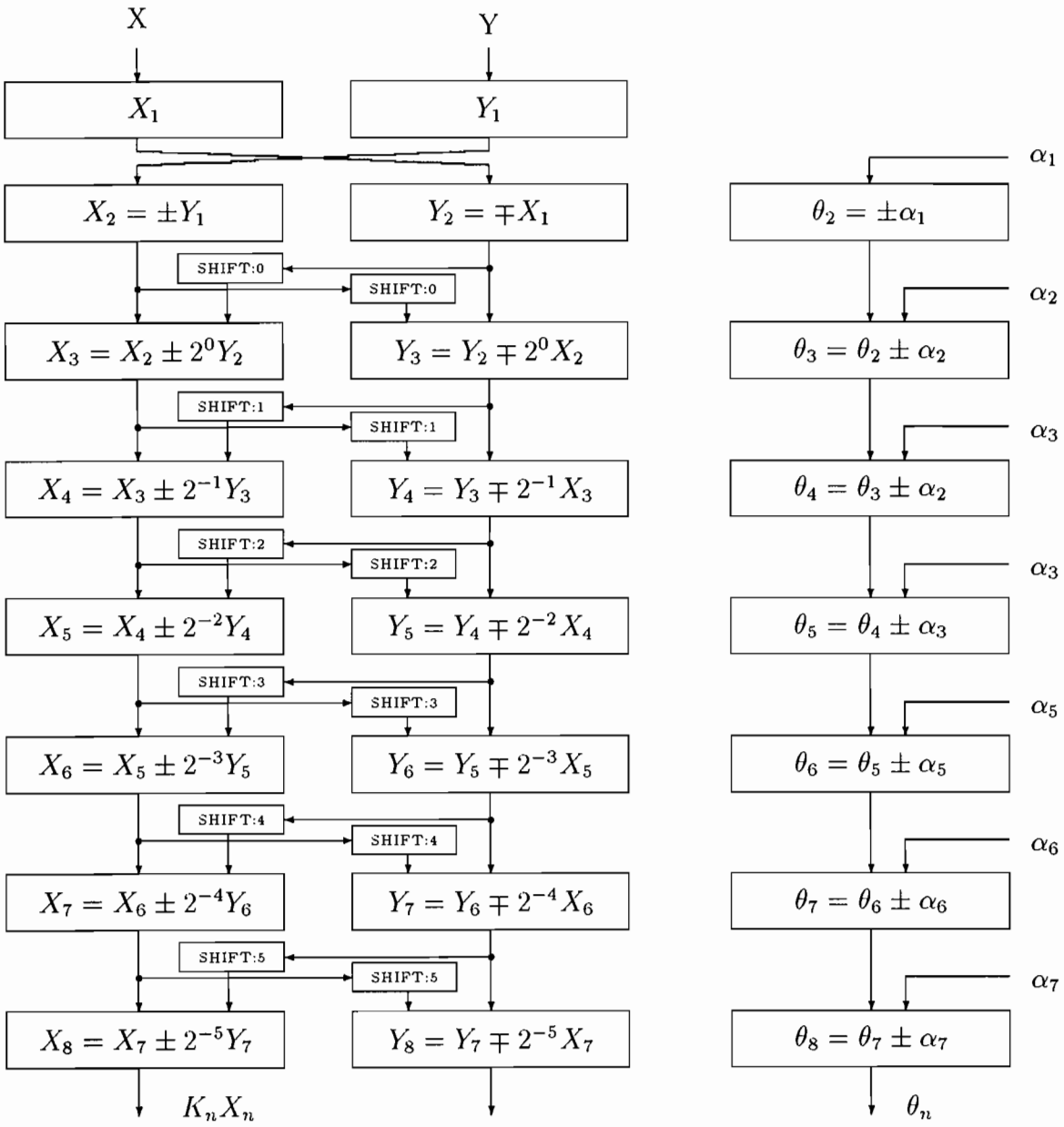


Bild 4.2.7 Realisierung des CORDIC-Algorithmus als vollständige Pipeline-Struktur.

4.2.4 Hilbert–Integer–Band–Aufteilung für BP–ΣΔ–Wandler

Bandpaß–ΣΔ–Wandler, Hilbert–Abtaster und der CORDIC–Algorithmus zusammen stellen eine effiziente Realisierung eines universellen Demodulators incl. schmalbandiger Selektion dar, siehe Bild 4.2.8. Es sei nun betrachtet, wie diese Prinzipien effizient miteinander kombiniert werden können.

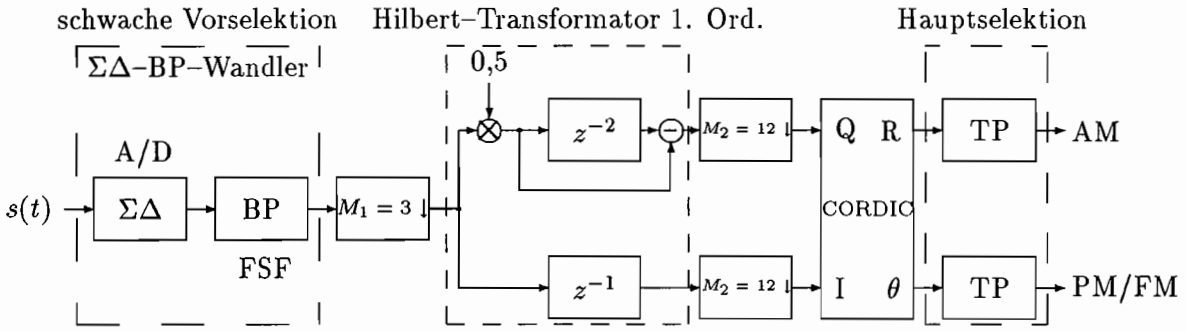


Bild 4.2.8 Hilbert–Transformator mit CORDIC–Demodulation bei $F = 12$ und $\downarrow M = 36$ entsprechend Tabelle 4.2.3.

Im Kapitel 3 wurde gezeigt, daß der Bandpaß–ΣΔ–Wandler eine effiziente Methode zur A/D–Wandlung incl. schmalbandiger Selektion ist. Als Filter wurden hier Frequenzabtastfilter vorgeschlagen, da sie besonders effizient realisiert werden können. Im letzten Abschnitt haben wird das CORDIC–Verfahren als Koordinatenwandler I,Q$\rightarrow R, \theta$ und damit unmittelbar einen AM– bzw. PM–Demodulator kennengelernt. Günstig für den CORDIC–Implementierungsaufwand war es, wenn die Bit–Breite der Eingangsworte des CORDIC–Prozessors relativ gering ist. Mit Hilfe des in Abschnitt 4.1.4 vorgestellten Hilbert–Transformators besteht nun die Möglichkeit, beim Bandpaß–ΣΔ–Wandler die Transformation I,Q$\rightarrow R, \theta$ vor der Hauptselektion durchzuführen, siehe Bild 4.2.8. Für den Aufwand der Hauptselektion ist es dabei unerheblich, ob zunächst I– und Q–Phase schmalbandig gefiltert werden und dann transformiert wird, oder ob zunächst transformiert wird und dann anschließend R und/oder θ gefiltert wird. Der Realisierungsaufwand für den CORDIC–Prozessor kann durch das Vorziehen erheblich reduziert werden. Weiterhin eröffnet sich bei früh im Signalpfad liegender Koordinatenwandlung die Möglichkeit, daß, falls nur die AM– oder PM/FM–Demodulation gewünscht ist, ein vollständiges Filter der Hauptselektion eingespart werden kann. Im folgenden sind einige Optimierungen für eine günstige Wahl der Parameter der FSF–Filter angegeben.
Es gelten für die Optimierung bei Hilbert–Aufteilung folgende Randbedingungen:

I) K–te und $(K+1)$–te Nullstelle der Elementarfilter sollen symmetrisch zur Signalfrequenz liegen $\rightarrow (K+0,5)/D = f_0/f_a = F$ mit $K \in \mathcal{N}$

II) Die Signalfrequenz sollte mit der Grenze eines Integer–Bandes zusammenfallen $\rightarrow l/M = f_0/f_a = 1/F$ mit $l \in \mathcal{N}$

III) Die selektierte Bandbreite sollte im Bereich 5-10 kHz liegen $\rightarrow f_a/D = 5 - 10$ kHz

IV) Die Multiplikationen mit $\cos(2\pi K/D)$ und $\cos(2\pi(K+1)/D)$ sind mit möglichst wenig Operationen im pseudoternären System auszuführen und die resultierende Abweichung der pseudoternären Addition sollte möglichst klein sein.

V) Die für die Hilbert–Transformation günstige $(T/4)$–Abtastung sollte möglich sein $(\rightarrow F \text{ modulo } 4) = 0$

Die Optimierung der Parameter M, K und D für 2 bis 6 pseudoternäre Operationen zur Realisierung der Funktionen $\cos(2\pi K/D)$ und $\cos(2\pi(K+1)/D)$ ergab dabei für $f_a = F \cdot 77{,}5$ kHz; $\Delta f = 5 - 20$ kHz; $M = 20 \cdots 80$ die Resultate aus der Tabelle 4.2.3.

F	D	K	M	Δf	effektive Bit bei		
					2 Op.	4 Op.	6 Op.
8	40	2	32	23250 Hz	5,97		
8	188	23	40	4947 Hz		13,02	20,44
12	188	15	36	5000 Hz	5,83	14,50	18,48
16	40	2	32	23250 Hz	5,97		
16	72	4	32	12917 Hz		17,72	22,95

Tabelle 4.2.3 Optimierung der FSF–Parameter bei zwei bis sechs pseudoternären Operationen und Hilbert–Bandaufteilung.

4.2.5 Vergleich verschiedener Demodulationsprinzipien

In [Wie88, S. 111-118] wurde vorgeschlagen, die Demodulation für AM–, PM– und FM–Signale durch Lösen der algebraischen Gleichungen (1.1.13) bis (1.1.16) mit einem Signalprozessor durchzuführen. Dieser Aufwand soll nun mit der Realisierung des CORDIC–Algorithmus in einem programmierbaren Gate Array verglichen werden. Dabei wird davon ausgegangen, daß der DSP Multiplikation, Addition und Schiebeoperationen in einem Zyklus ausführt. Da die algebraischen Gleichungen nicht unmittelbar berechenbar sind, muß eine Taylor–Reihenentwicklung um x_0 benutzt [BS84, S. 269] werden:

$$f(x) = \sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!}(x - x_0)^i + R_n(x) \tag{4.2.15}$$

wobei

$$R_n(x) = \frac{f^{(n+1)}(x_0 + \lambda(x - x_0))}{(n+1)!}(x - x_0)^{n+1} \tag{4.2.16}$$

mit $\lambda \in (0, 1)$.

Eine konventionelle DSP–Lösung ist dann vorteilhaft, wenn die zu berechnende Funktion in einer (kurzen) Taylor–Reihe entwickelbar ist und so der schnelle Hardware–Multiplizierer des DSPs vorteilhaft eingesetzt werden kann. FIR–Filter und Auto– bzw. Kreuzkorrelationen sind Beispiele eines effizienten Einsatzes eines DSPs. Das DSP–Konzept ist

aber nur schlecht geeignet, wenn Division oder die Berechnung trigonometrischer Funktionen nötig sind. So konvergiert die Taylor–Reihe der arctan–Funktion nur linear mit der Anzahl der Iterationen [BS84, S. 34], d. h. bei einer Verdoppelung der Iterationszahl gewinnt man ein Bit an Genauigkeit — anders hingegen bei dem CORDIC–Algorithmus, wo pro Iteration ca. ein Bit an Auflösung gewonnen wird. Eine detaillierte Untersuchung der notwendigen Rechenzyklen wurde in [MB91, S. 33-37] durchgeführt.

Der Vergleich zwischen algebraischer Berechnung mittels DSP und CORDIC–FPGA--Realisierung als universelles Demodulationsprinzip läßt sich folgendermaßen zusammenfassen: Die direkte algebraische Umsetzung der I/Q–Gleichungen (1.1.13) bis (1.1.16) zur Demodulation von AM–, PM– und FM–Signalen mit einem DSP bereiten vor allem Schwierigkeiten bei PM–, weniger bei FM– und AM–Signalen. Der CORDIC–Algorithmus hingegen benötigt bei der iterativen Verarbeitung von b-Bit Worten nur ca. b Zyklen, und es ist möglich eine vollständige Pipeline–Struktur zu realisieren, so daß, abgesehen von einer konstanten Verzögerung, eine Wandlung I/Q nach R/θ in einem Zyklus erfolgen kann (AM– bzw. PM–Demodulation). Beim FM–Signal muß nach der CORDIC–Wandlung lediglich *eine* Differenzierung erfolgen.

Abgesehen von zweistufigen ASK–Signalen zeigt sich, daß die CORDIC–Realisierung mit Hilfe eines programmierbaren Gate Arrays bei allen Modulationsarten der direkten algebraischen Berechnung mit dem DSP vorzuziehen ist, da diese Lösung erhebliche Rechen– bzw. Geschwindigkeitsvorteile besitzt.

4.3 Optimalstrategien zur inkohärenten FM–Demodulation

Um verschiedene FM–Demodulatoren objektiv vergleichen zu können, ist es zunächst nötig, das im Kapitel 1 vorgestellte Signalmodell für die FM–Modulation zu verfeinern. Mit digitale FM–Modulation (dFM) bezeichnet man dabei das Verfahren zur Übertragung von **digitalen** Signalen über eine (analoge) FM–Strecke. Das Bild 4.3.1 zeigt das verwendete Modell der dFM–Übertragungsstrecke.

Ein zweistufiger Sendevektor $\vec{\alpha}$ gelangt auf einen Impulsgeber, an den sich ein Formfilter mit der Impulsantwort des Frequenzgrundimpulses (FGI) anschließt. Nach der Multiplikation mit dem 2π–fachen des Modulationsindex h gelangt dieses Signal auf den FM–Modulator, der eine Mittenfrequenz ω_0 und eine zufällige Anfangsphase ϕ_0 besitzt. Die Momentanfrequenz des Ausgangssignals des Modulators ergibt sich damit zu

$$\omega(t) = \frac{\mathrm{d}\,\phi(\vec{\alpha},t)}{\mathrm{d}\,t} \tag{4.3.1}$$

$$= \omega_0 + 2\pi h \sum_{i=0}^{n} \alpha_i g(t-iT), \tag{4.3.2}$$

wobei α_i aus dem M stufigen Alphabet: $\pm 1, \pm 3 \ldots \pm M$ gewählt ist und für das gesamte Ausgangssignal des Modulators folgt

$$s(t) = \sqrt{2P}\cos\left[\omega_0 t + 2\pi h \sum_{i=0}^{n} \alpha_i q(t-iT) + \phi_0\right] \tag{4.3.3}$$

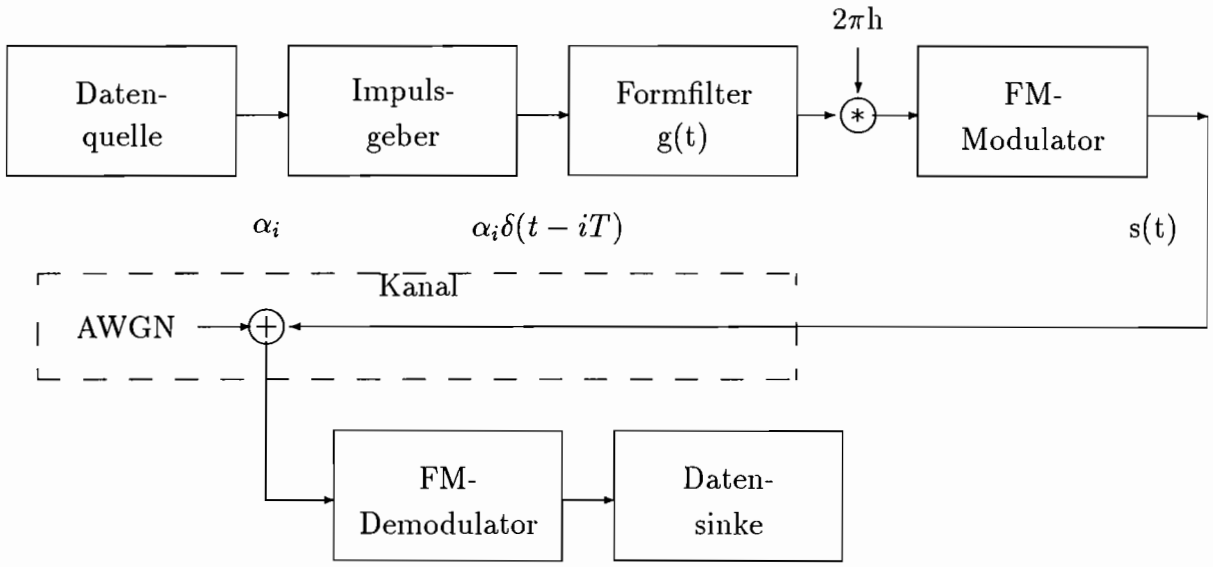


Bild 4.3.1 Typische Darstellung einer Übertragungsstrecke für die digitale FM–Modulation.

wobei $g(t) = \frac{d\,q(t)}{d\,t}$ gilt.

Üblich sind kausale Frequenzgrundimpulse mit endlicher Dauer von L Bit–Intervallen. Gilt $L = 1$, so spricht man von „Full Response" Frequenzgrundimpulsen, da die gesamte Auswirkung des Symbols $\alpha_i g(t - iT)$ innerhalb *eines* Bit–Intervalls liegt. Ist $L \geq 2$ so bezeichnet man diesen FGI mit „Partial Response" Signal†. Bild 4.3.2 zeigt nochmals die bei LW–Übertragung üblichen Frequenz– und Phasengrundimpulse.
Die zwei wichtigsten Parameter einer digitalen Übertragung sind:

- Bitfehlerwahrscheinlichkeit (Störanfälligkeit bei gegebenem $\frac{S}{N}$).

† Eine „Partial Response" FGI wird bei der Übertragung im Ton/Datenteil der D2MAC Satellitenfernsehnorm benutzt [Gro91].

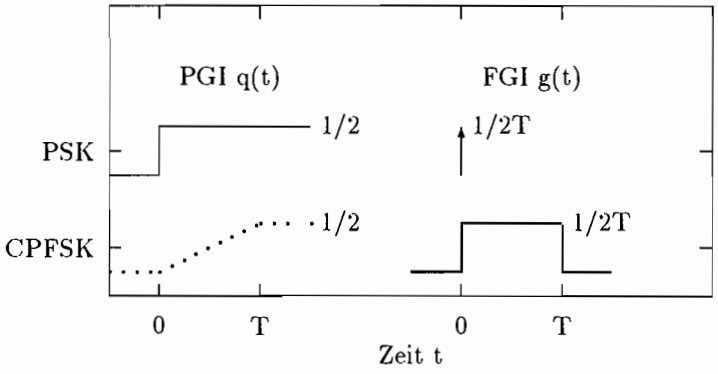


Bild 4.3.2 Phasengrundimpuls (linke Spalte) und Frequenzgrundimpuls (rechte Spalte) einiger dFM–Modulationsverfahren [AAS86, S. 52].

- belegte spektrale Bandbreite.

Für die belegte spektrale Bandbreite wurden bereits in Abschnitt 1.2 Simulationsergebnisse angegeben. Die Bestimmung der Bitfehlerwahrscheinlichkeit (BFW) gelingt über die Berechnung der *minimalen Euklidischen Distanz* (siehe Abschnitt 4.3.1). Es wird sich zeigen, daß mittels der minimalen Euklidischen Distanz effizient verschiedene Empfänger miteinander verglichen werden können.

4.3.1 Abschätzung zur Bitfehlerwahrscheinlichkeit

Die Berechnung der BFW gelingt, vor allem bei großen Werten von $\frac{S}{N}$, sehr gut mit Hilfe folgender Abschätzung [AAS86, S. 20-22]

$$P_e \sim Q\left(d_{min}\{s_i(t), s_k(t)\} \cdot \sqrt{E/N_0}\right). \tag{4.3.4}$$

wobei $Q(x)$ die Fehlerfunktion ist

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-\nu^2}{2}}\, dt = \frac{1}{2} erfc\left(\frac{x}{\sqrt{2}}\right). \tag{4.3.5}$$

$s_i(t)$ und $s_k(t)$ sind zwei Signale entsprechend Gleichung (4.3.3), die in N Bit–Intervallen differieren. Normiert man $||s_i(t) - s_k(t)||^2$ auf $2PT$, so wird dies als Quadrat der **Euklidischer Distanz** (ED) bezeichnet.

Der **Fehlervektor** $\vec{\gamma}$ ist die Differenz zwischen der Sendefolge $\vec{\alpha}$ des Signals $s_i(t)$ und der geschätzten Sequenz $\hat{\vec{\alpha}}$ des Signals $s_k(t)$.

$$\gamma_i = \alpha_i - \hat{\alpha}_i \qquad\qquad \gamma_i \in [-2, 0, 2] \tag{4.3.6}$$

Die Differenzsequenz $\vec{\gamma} = \vec{\alpha} - \hat{\vec{\alpha}}$ mit dem geringsten Wert der ED wird als **minimale Euklidische Distanz** (d_{min}) bezeichnet und liefert den größten Beitrag zur BFW, da $P_e \sim e^{-d_{min}}$ gilt. Unter der Voraussetzung, daß die Modulationsfrequenz sehr viel größer als der Frequenzhub πh ist, folgt für

$$\int \cos(\omega_0 + \phi_i(t)) \cdot \cos(\omega_0 + \phi_k(t))\, dt = \int \cos(\phi_i(t) - \phi_k(t))\, dt. \tag{4.3.7}$$

Somit ergibt sich für die Euklidische Distanz

$$d^2 = \frac{1}{2PT}||s_i(t) - s_k(t)||^2 = \frac{1}{T}\int_0^{NT} 1 - \cos\left(\phi(\vec{\gamma}, t)\right)\, dt \tag{4.3.8}$$

mit

$$\phi(\vec{\gamma}, t) = 2\pi h \sum_{\nu=0}^{M} \gamma_\nu q(t - \nu T). \tag{4.3.9}$$

Die obere Schranke der minimalen Euklidischen Distanz wird mit Hilfe des M-ten Merger (Vernichters) bestimmt. Die Länge der Fehlersequenz beträgt M Bit–Intervalle, so daß nach der Zeit $t = (L + M)T$ die Fehlerphase $\phi_i(t) - \phi_k(t)$ Null wird. Für die Fehlersequenz $\vec{\gamma}$ folgt deshalb

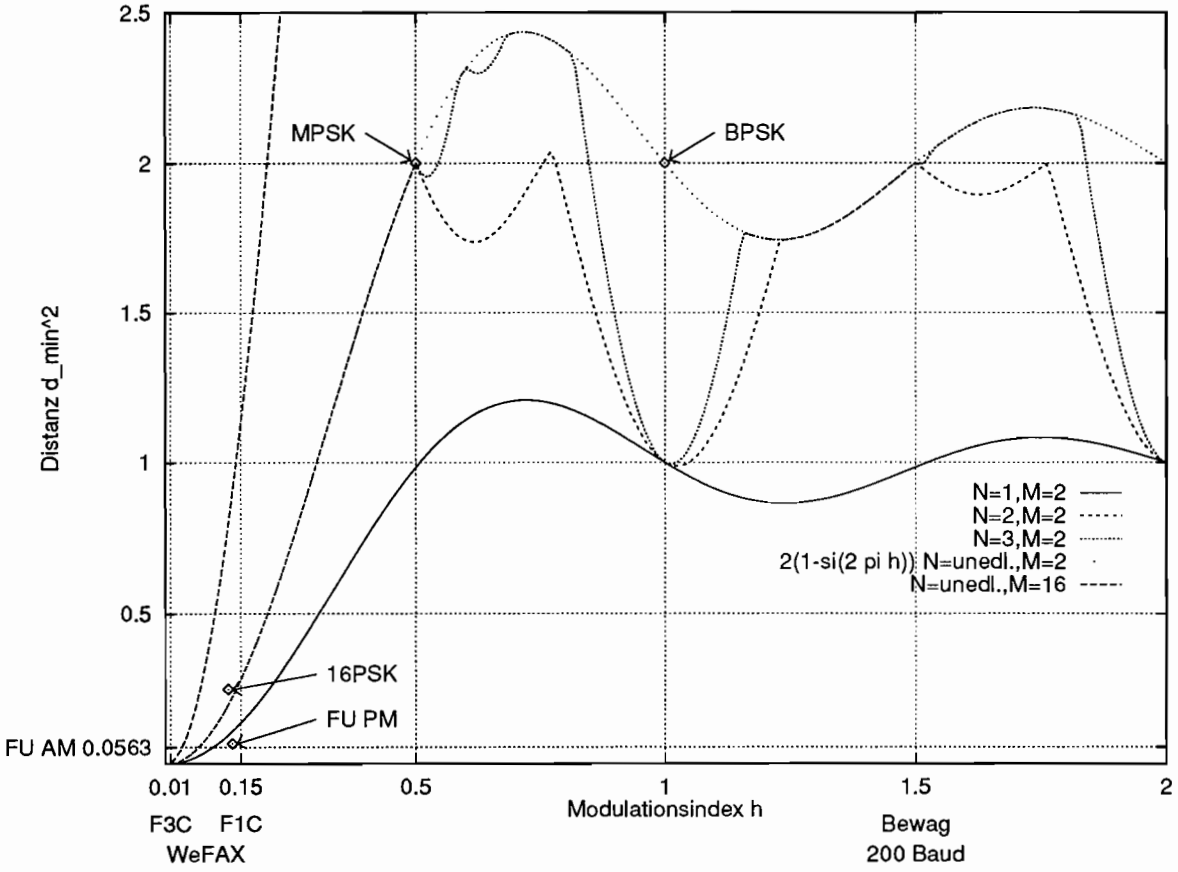


Bild 4.3.3 Die minimale Euklidischen Distanz der verschiedener LW–Datensignalen.

$$\gamma_i \in \begin{cases} \{2,0,-2\} & 0 \le i \le M \\ \{0\} & i < 0, i > M \end{cases} \qquad \sum_{i=0}^{M} \gamma_i = 0 \tag{4.3.10}$$

Da $\vec{\gamma}$ und $-\vec{\gamma}$ die gleiche ED besitzen, kann man ohne Einschränkung der Allgemeinheit $\gamma_0 = 2$ ansetzen. Weiterhin läßt sich zeigen, daß nur die $M \le L$–Merger zur Abschätzung untersucht werden müssen. Da bei LW–Signalen nur „Full Response"–Signale Verwendung finden, beschränkt sich somit die Analyse auf den ersten „Merger".

In Bild 4.3.3 sind die Minimalen Euklidischen Distanzen d^2_{min} verschiedener Modulationsarten über den Modulationsindex h aufgetragen. Als Referenzpunkte sind BPSK und MPSK mit eingezeichnet, die für Empfänger, die nur ein Bit auswerten (Gedächtnislänge $N = 1$, siehe Abschnitt 4.3.3), optimale Ergebnisse liefern [AAS86, S. 92]. Für den REC–Frequenzgrundimpuls des CPFSF–Signals (siehe Bild 4.3.2) ergibt sich für den Fall $L = 1$ die maximal erreichbare Distanz (Gedächtnislänge $N = \infty$) zu $d^2_B = 2(1 + \mathrm{si}(2\pi h)$, und damit ergibt sich als optimaler Wert $h_{opt} = \frac{4{,}49341}{2\pi} = 0{,}71515$, der oberhalb der Distanzen von BPSK und MPSK liegt. Weiterhin ist dargestellt, welche Distanzen sich nach endlich vielen Bit–Intervallen berechnen. Die obere Schranke d^2_B wird bereits für $N = 2$ erreicht. Außerdem wurde versucht, die für WeFAX und Bewag gewählten Modulationen ebenso wie die Distanzen für Funkuhrmodulation (PM– und AM–FU) entsprechend den

Berechnungen aus [Wie88, S. 70-78] mit in das Diagramm aufzunehmen. Die für Funkuhren gewählte Modulationsart bleibt weit unter dem zurück, was mittels LW Ausstrahlung möglich wäre, da die Verwendung als reiner Normalfrequenzsender sonst zu stark gestört würde. Die FM–Modulationen (WeFAX und Bewag) ergeben bei Berücksichtigung von mindestens 2 Bit–Intervallen für die gewählte Baud–Rate und 1REC–Modulation optimale Werte. Der Vergleich der Distanzen für $M = 2$ und $M = 16$ zeigt, daß das mehrstufige Verfahren bei gleichem maximalem Phasenhub (0,15) erheblich störempfindlicher wird als das F1C–Signal.

4.3.2 Strategien zur Optimaldemodulation für FM–LW–Signale

Es ist aus der Literatur bekannt [Lü88, S. 134], [Skl88, S. 88], [Pap86, S. 298-300], daß der Matched–Filter–Empfänger derjenige Empfänger ist, der für einen durch additives Gaußsches Rauschen gestörten (AWGN) Kanal ein Maximum des S/N–Verhältnisses am Demodulatorausgang bewirkt. Bild 4.3.4 zeigt die Realisierung der Korrelationsempfängers [OL74] und eine Matched–Filter–Anordnung für F1C modulierte Signale. Für das Korrelationsfilter gilt $h(t) = s_0(t) - s_1(t)$ und $\otimes$ symbolisiert die Faltungsoperation.

In dieser wie auch den weiteren Betrachtungen in diesem Kapitel wird von (nahezu) idealer Bit–Synchronisation ausgegangen, die im allgemeinen bei der dFM erst bei hohen S/N–Verhältnissen erreicht wird [SW92], [ACM82]. Bei der LW–FAX–Übertragung ist aber keine Bit– bzw. Pixel–Synchronisation vorhanden, so daß die Bit–Synchronisation, obwohl sie im allgemeinen bei der dFM sehr wichtig ist und nicht vernachlässigt werden sollte, nicht weiter betrachtet wird [Fra92], [Gar88].

Wird zunächst eine Koordinatentransformation $I, Q \rightarrow R, \theta$ durchgeführt und wird dann nur die Phase θ für die PM/FM–Demodulation verwendet, so werden im allgemeinen geringfügig schlechtere Ergebnisse erwartet, da nur die Phase für den Korrelationsempfang benutzt wird. Hingegen bei der sin/cos–Korrelation mit I– und Q–Phase wird implizit die Amplitude „R“ auch ausgewertet. Vorteilhaft ist beim Phasen–Korrelationsempfang, daß das Optimalfilter für die Phase sich als einfache Differenz von Anfangsphasenwert minus Endphasenwert darstellt ($\int_0^T g(t)dt = q(T) - q(0)$, $I\&D$–Demodulator), so daß

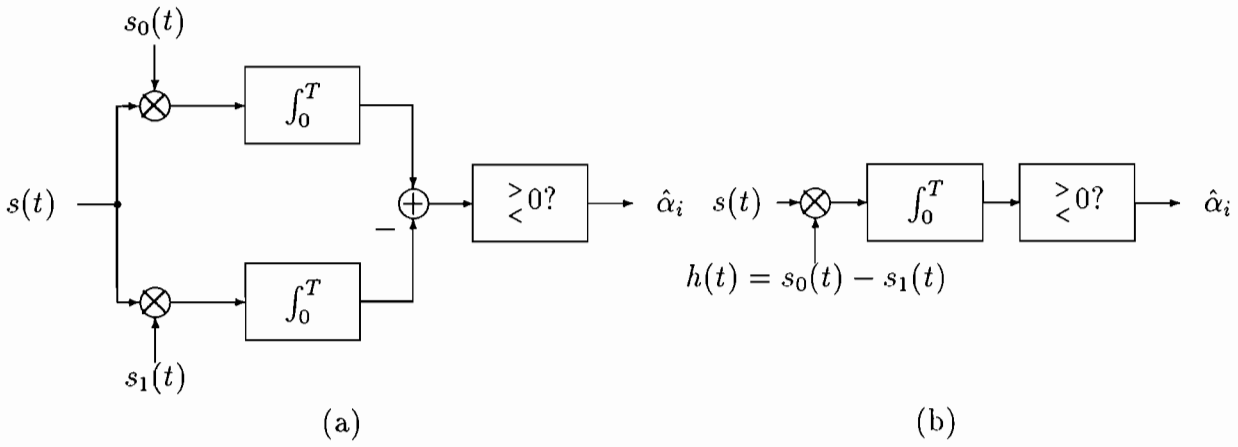


Bild 4.3.4 (a) Korrelationsempfänger. (b) Matched–Filter–Empfänger.

letztlich nur *eine* Wandlung I,Q→ R, θ pro Bildpunkt für die fortlaufende optimale FM-Demodulation mittels Phase nötig ist. Allerdings erscheint diese Lösung anfällig für impulsförmige Störungen im Bereich $q(T)$ oder $q(0)$.

Weniger anfällig bei Impulsstörungen erscheint die Demodulation mittels Geradenapproximation mehrere Phasenwerte eines Bildpunktes nach dem Verfahren der kleinsten Quadrate (LMS-Geraden) zu sein. Die Phasenwerte θ bei der FM-Demodulation zeigen innerhalb eines Bildpunktes einen linearen Verlauf, so daß die *Steigung* der Geraden proportional dem Sendesymbol α_i ist.

Als vierte Alternative wurde eine Mittelung über die Differenz benachbarter Phasenwerte untersucht. Hierbei ist es vorteilhaft, daß man für die maximal zulässige (betragsmäßige) Phasendifferenz eine Schranke einführen kann, so daß wiederum Impulsstörungen herausgefiltert werden können.

Im folgenden sollen für einen AWGN-Kanal Simulationsergebnisse dieser Prinzipien diskutiert werden‡. Die Verfahren „LMS-Geraden" und „Mittelung" wurden zur Verbesserung der Demodulationsergenisse in dem „Hilbert-Abtastempfänger mit CORDIC-Demodulation" [Sch92] implementiert.

Die allgemeine Aussage der Optimalität des sin/cos-Korrelationsempfangs ist bei einem realen System nicht mehr immer gültig. So zeigen die Bilder 4.3.5(a) und (b) die Bandbreitenoptimierung für WeFAX und Bewag LW-Signale. Deutlich wird, daß für sehr breitbandige Filterungen $f_g > 2f_s$ der sin/cos-Korrelationsempfänger optimal ist — hingegen bei der günstigen schmalbandigen Filterung erweist sich der $I\&D$-Demodulator beim WeFAX-Signal als vorteilhaft. Hier ist besonders interessant, daß mit einer geeigneter Filterung sich bei der Berechnung der zu der Fehlerwahrscheinlichkeit gehörenden minimalen Euklidischen Distanz $d_{\min} = 0,106$ ein ähnlich günstiger Wert ergibt, wie für die theoretische Obergrenze von $2(1 - \sin(2\pi h)/(2\pi h)) = 0,283$. Bei speziellen Modulationsindizes und optimaler Filterung traten sogar Werte auf, welche die theoretische Obergrenze d_B^2 übertrafen. Dieser interessante Sachverhalt ist in [AAS86, S. 131] bereits als Möglichkeit erwähnt. Plausibel wird dies durch folgende Überlegung: Durch die Fil-

‡ Bei der Simulation mit einem AWGN-Kanal ist das für impulsförmige Störungen günstige Verhalten der Demodulatoren „LMS-Geraden" und „Mittelung über Phasendifferenzwerte" natürlich nicht zu beobachten.

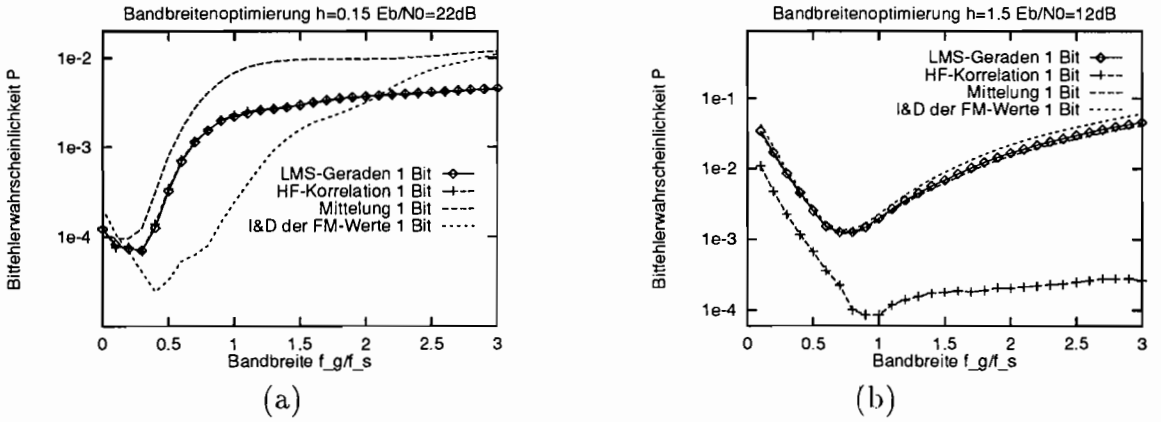


Bild 4.3.5 Bandbreitenoptimierung bei Übergangsbandbreite $f_ü = f_s$ für den (a) WeFAX-Empfänger $h = 0,15$ und (b) Bewag-Empfänger $h = 1,5$.

terung entsteht für die minimale Differenzsequenz ein größerer Wert, wie sich z. B. auch bei RC–FGI eine größere Distanz entwickelt als bei dem 1REC–FGI. Durch die schmalbandige Filterung wird der Phasenverlauf so stark verändert, daß man im Prinzip gar nicht mehr von einem 1REC–Phasenverlauf und den zugehörigen oberen Schranke für die minimale Euklidische Distanz d_B^2 ausgehen darf.

4.3.3 Gedächtnisbehaftete Demodulation

Anders als bei AM– oder PM–Demodulation ist bei der FM–Demodulation die Berücksichtigung benachbarter Bit–Intervalle zur Entscheidung des mittleren Bits vorteilhaft. Dies läßt sich im Phasendiagramm folgendermaßen veranschaulichen: Bestimmt man die optimalen Geraden (LMS–Demodulator siehe Gleichungen (4.3.11) bis (4.3.15)) für $\hat{\alpha}_{i-1}$ und $\hat{\alpha}_{i+1}$, so kann aus dem Endpunkt von $\hat{\alpha}_{i-1}$ und dem Anfangspunkt von der zu $\hat{\alpha}_{i+1}$ gehörenden Geraden auf das zu decodierende $\hat{\alpha}_i$ geschlossen werden. D. h. trotz einer starken Störung in einem Bereich, die bei Berücksichtigung von nur einem Bit–Intervall zu einer Fehlentscheidung führt, kann mittels einer „Gerade mit Knickpunkten am Bit–Intervallende“ von z. B. 3 Bit Länge doch noch eine richtige Decodierung erfolgen. In Bild 4.3.6 ist das Ergebnis einer Simulation für die AWGN–Strecke aus Bild 4.3.1 bei Verwendung von LMS–Geraden der Gedächtnislänge $N = 1$ und $N = 3$ gezeigt. Zu erkennen ist eine starke Störung in Bereich des zu decodierenden Bits $\hat{\alpha}_i$, die bei Berücksichtigung des Phasenverlaufs von nur einem Intervall (Linie —) zu einer Falschentscheidung geführt hätte. Die Berücksichtigung der benachbarten Intervalle (Linie •–) bewirkt hingegen, daß die korrekte Entscheidung $\hat{\alpha}_i = -1$ getroffen wird.

Allerdings steigt der Berechnungsaufwand exponentiell mit der Länge der betrachteten Bit–Intervalls ($\sim 2^N$). Laut Diagramm 4.3.4 ist eine Beobachtung von $N = 2$ Bit–Intervallen bereits ausreichend, um die oberere Schranke der Euklidischen Distanz d_B^2 für $h = 1,5$ bzw. $h = 0,15$ zu erreichen. Da aber, wie in Bild 4.3.8 noch gezeigt wird, das zu decodierende Bit in der Mitte liegen muß, ist es nur sinnvoll, ungerade Gedächt-

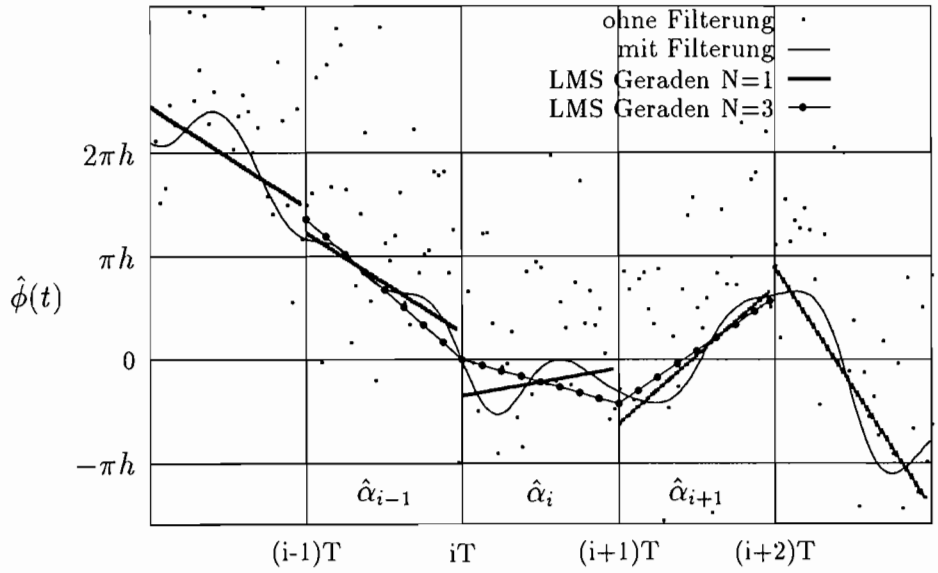


Bild 4.3.6 Demodulation mittels LMS–Geraden über drei Bit.

nislängen zu verwenden. Simulationen ergaben, daß bei der Decodierung des ersten oder zweiten Bits bei $N = 2$ eine wesentliche Verschlechterung gegenüber einer Gedächtnislänge $N = 3$ auftritt. Würde man das zu beobachtende Intervall in die Mitte legen, so müßte jeweils 1/2 Bit vor und nach dem zu decodierenden Bit auch berücksichtigt werden, d. h. es müssen trotzdem $2^3 = 8$ verschiedene sin/cos–Korrelationen durchgeführt werden, so daß es einfacher erscheint, direkt mit einer Gedächtnislänge $N = 3$ zu arbeiten. Bei vollständiger Berücksichtigung des Gedächtnisses von $N = 3$ müssen hier ebenso nur $2^3 = 8$ verschiedene sin/cos–Korrelationen durchgeführt werden, wobei allerdings bei T Abtastwerten pro Bit–Intervall für jede Korrelation T Multiplikationen mehr auftreten als bei $N = 2$.

Ist die Anfangsphase bekannt und liegt ein rationaler Modulationindex $h = 2K/D$ vor, so kann man noch einen Vorteil dadurch erhalten, daß der Phasenverlauf immer modulo 2π berechnet wird. Man geht dann von der Berechnung aller 2^N–Kombinationen (Maximum Likelihood Estimation Empfänger [AAS86, S. 241]) ab und verwendet einen Sequenzschätzer (MLSE, Viterbi–Algorithmus) zur Bestimmung der Phasenpfades mit der größten Wahrscheinlichkeit. Dies ist deshalb nützlich, da so nur die D wahrscheinlichsten Pfade im Phasen–Trellis§ weiterverfolgt werden müssen. Dies führt bei großen Gedächtnislängen zu einer erheblichen Auswandsreduktion. Die Anwendung des MLSE–Empfängers auf inkohärente Demodulation ist möglich, führt aber zu einem erheblichen Verlust [MB89, S. 41], [Gro91].

Bei kleiner Gedächtnislänge und kleinem Modulationsindex d. h. großes D (WeFAX) aber bringt die MLSE–Berechnung mittels Viterbi–Algorithmus keinen Vorteil, da hier noch keine „Doppelbelegung“ von Knoten im Trellisdiagramm vorkommt. Für Bewag–Modulation ist ein geringer Berechnungvorteil beim $N = 3$ Viterbi–Algorithmus vorhanden.

Ein weitaus größerer Berechnungvorteil insbesondere bei mehrstufiger (z. B. $M = 16$) Übertragung ergibt sich bei der Realisierung einer Demodulation mit LMS–Geraden. Statt der $M \cdot 2^N$ verschiedenen Korrelationen können hier auf direktem Weg die LMS–Geraden berechnet werden. Die Schätzfunktion $\hat{\phi}[n]$ für den Phasenverlauf berechnet sich aus den einzelnen Phasenwerten $\phi[n]$ nach dem LMS–Verfahren folgendermaßen [SH90, S. 11-15]:

$$\hat{\phi}[n] = \sum_{i=0}^{N} c_i \phi_i[n] \tag{4.3.11}$$

$$\text{wobei} \quad C = \Phi^{-1} \cdot F \tag{4.3.12}$$

$$\Phi_{i,j} = \sum_{n=0}^{N \cdot T} \phi_i[n] \cdot \phi_j[n] \tag{4.3.13}$$

$$F_i = \sum_{n=0}^{N \cdot T} c_i \phi[n] \tag{4.3.14}$$

§ Zum Trellis–Diagramm siehe auch die entsprechende Darstellung bei Faltungscodes in Bild 5.2.6.

$$\phi_i[n] = \begin{cases} 0 & : N \cdot T \geq n \leq iT \\ 1 & : i \cdot T \leq n \geq N \cdot T \\ \frac{n-(i-1)\cdot T}{T} & : (i-1) \cdot T \leq n \geq i \cdot T \forall i \geq 1 \end{cases} \quad (4.3.15)$$

Dabei ist jedes Bit–Intervall durch T Abtastwerte dargestellt. Für $N = 3$ zeigt das Bild 4.3.7(a) eine Veranschaulichung der gewählten Approximationsfunktionen $\phi_i[n]$. Dabei sind die Approximationsfunktionen $\phi_i[n]$ so gewählt worden, daß c_0 den Gleichanteil der Knick–Gerade beschreibt, während die $c_i \forall i \geq 1$ unmittelbar die Steigung der jeweiligen Geradenstücke angeben. Somit folgt für die gesuchte Steigung des mittleren Bildpunkts bei $N = 3$ Bit:

$$c_2 = \Phi_{31}^{-1} F_0 + \Phi_{32}^{-1} F_1 + \Phi_{33}^{-1} F_2 + \Phi_{34}^{-1} F_3 \quad (4.3.16)$$

Vorteilhaft ist, daß die Matrix Φ^{-1} nur von den Musterfunktionen $\phi_i[n]$ abhängt und somit nur einmal berechnet werden muß und die Φ_{3j}^{-1} in eine sehr kleinen Tabelle passen. Bei einer fortlaufenden Berechnung ist es vorteilhaft, zunächst intervallweise eine Korrelation mit der linear ansteigenden Gerade und Summation über die T Werte durchzuführen (zwei Metrik–Werte pro Bit–Intervall). Dann kann die Bestimmung der F_i durch Summation jeweils von N Werten entsprechend dem Schema aus Bild 4.3.7(a) erreicht werden. Dem Bild 4.3.7(b) entnehmen wir, daß sich der Rechenaufwand durch die Metrik–Berechnung nochmals erheblich reduziert. Dabei ist von Vorteil, daß die Demodulation mittels LMS–Geraden unabhängig von der Anzahl der Graustufen ist, d. h. für $M = 2$ und $M = 16$ ergibt sich der gleiche Rechenaufwand. Bei der sin/cos–Korrelation hingegen ergibt sich ein achtfacher Rechenaufwand bei Berücksichtigung von 16 Graustufen. Allerdings muß berücksichtigt werden, daß z. B. für das Bewag–Signal der sin/cos–Korrelationsempfänger bei AWGN–Störung erheblich bessere Ergebnisse liefert als die Demodulation mittels LMS–Geraden, siehe Bild 4.3.5, so daß der erhöhte Aufwand in diesem Fall gerechtfertigt erscheint.

Dieser 3 Bit Phasenkorrelationsdemodulator mit LMS–Geraden wurde, wie erwähnt, im Empfänger [Sch92] realisiert und lieferte merkliche Verbesserungen. Dies wird auch

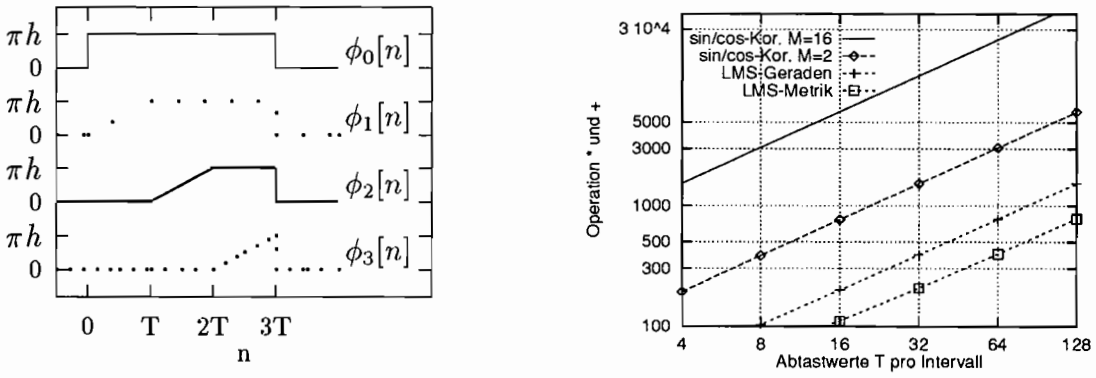
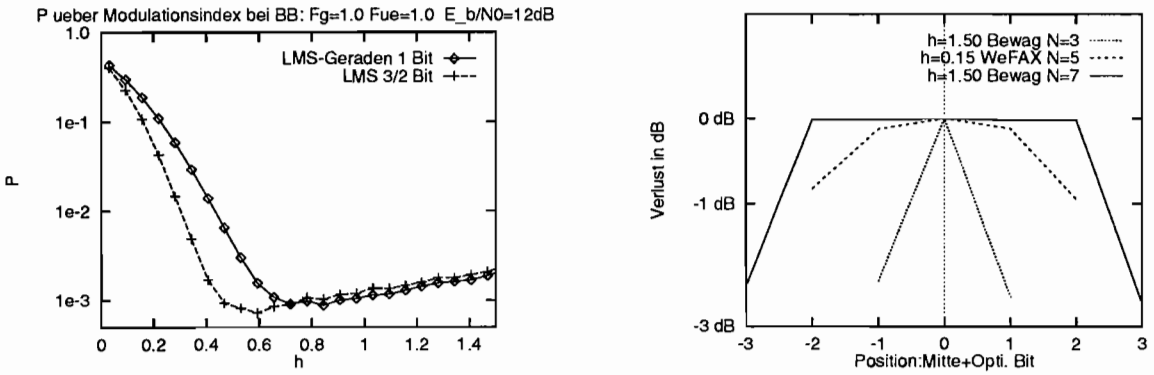


Bild 4.3.7 (a) Musterfunktionen für die LMS–Geradenapproximation über drei Bit. (b) Berechnungsvorteil durch LMS–Geradenapproximation im Vergleich zur sin/cos–Korrelationsempfang für einen DSP.



(a) (b)

Bild 4.3.8 (a) Vergleich der Demodulation von LMS–Geraden bei Gedächtnislänge/zu decodierendes Bit: 1/1 und 3/2. (b) Optimale Position des zu decodierenden Bits.

aus den Simulationsergebnissen in Bild 4.3.8(a) deutlich. Für einen Modulationsindex $h \leq 0,7$ bringt die Demodulation mit LMS–Geraden der Länge $N = 3$ bei Decodieren des mittleren Bits erhebliche Verbesserung. Für das WeFAX–Signal ($h = 0,15$) ergeben sich z. B. 2,95 dB Gewinn gegenüber der Demodulation mit $N = 1$. Für $h \geq 0,7$ ist die gewählte Filterung zu schmalbandig, so daß hier sogar die Ergebnisse für Gedächtnislänge $N = 1$ besser ausfallen als für $N = 3$.

Zum Abschluß sei noch in Bild 4.3.8(b) gezeigt, daß bei „Full Response“ Signalen wie das 1REC–WeFAX bzw. Bewag–FSK immer das mittlere Bit zur Decodierung benutzt werden sollte. Ein Verlust von bis zu 3 dB tritt auf, wenn ein anderes Bit als das mittlere zur Decodierung benutzt wird.

4.4 Kohärente Empfänger für Bandpaßsignale

Bereits im Abschnitt 1.1.2 wurde das Prinzip einer kohärenten Demodulation kurz beschrieben: Ein synchroner bzw. kohärenter Demodulator führt mittels einer Regelschleife einen Oszillator (VCO) dem Eingangssignal nach, wobei der Gleichanteil dieses Regelsignals unmittelbar das FM–demodulierte Signal darstellt, siehe Bild 4.4.1 und 4.4.2. Die beim universellen Empfänger geforderte PM–Demodulation erreicht man durch Integration des Regelsignals und die AM–Demodulation kann durch Mischen mit einem um 90° versetzten Oszillator mit anschließender TP–Filterung (Q–Zweig bei Bild 4.4.2) erreicht werden.
Als grundlegende Realisierungsformen unterscheidet man die zwei Typen

- Phasenregelschleife (Phase–Locked Loop, PLL)
- Costas–Regelschleife (Costas Loop, CL)

Im Bild 4.4.1 und Bild 4.4.2 sind PLL und CL mit deren wichtigsten Parametern dargestellt. Zu erkennen ist der nahezu doppelt so große Aufwand bei der Realisierung der Costas–Regelschleife gegenüber der PLL.

Beide Regelschleifen können analog (lineare PLL/CL) oder auch vollständig digital (ADPLL/ADCL) aufgebaut werden. Auch Mischformen wie „digitale PLL mit analo-

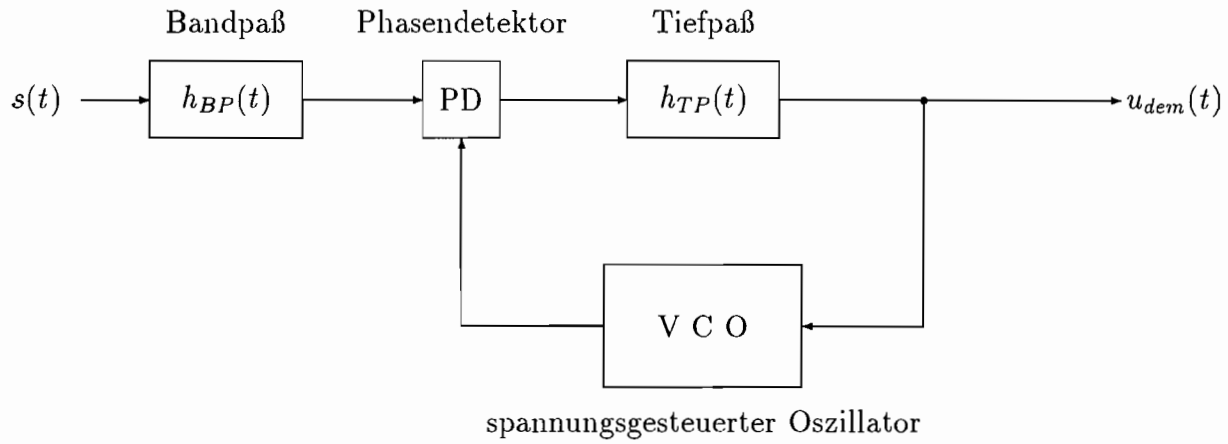


Bild 4.4.1 Phasenregelkreis (PLL) mit (obligatorischem) Eingangsbandpaß ($h_{BP}(t)$), bestehend aus: Phasendetektor (PD), Tiefpaß ($h_{TP}(t)$) und spannungsgesteuertem Oszillator (VCO).

gen Zwischensignalen" (DPLL) sind gebräuchlich und werden in Lehrbüchern ausführlich diskutiert, siehe z. B. [Vit66], [Gar79], [Ges84] oder [Bes87].

Da PLL und CL Regelschleifen sind, befassen sich Veröffentlichungen zu diesem Themenkreis weniger mit dem Verhalten der Regelschleifen bei speziellen Eingangssignalen [Osb82] oder mit speziellen Störspektren [Dav82], [Wei85], sondern meist mit dem dynamischen Stabilitätsverhalten [LC81], [SR64], [Ste88], [MT82], [MT81]. Die im folgenden aufgezeigten Untersuchungen in [Bra90], [GS93], [RS93] und [Sch91a] befassen sich mit Realisierungsaspekten von ADPLL und ADCL. Dabei wird sich zeigen, daß, wie vermutet, die *Signalselektion* zu aufwendigeren Realisierungen führt als dies bei den im Abschnitt 4.1.4 beschriebenen inkohärenten Empfängern der Fall ist. Die notwendigen Schaltungsteile für die kohärente *Demodulation* benötigen hingegen wesentlich weniger Platz in einem FPGA als die inkohärenten Demodulatoren.

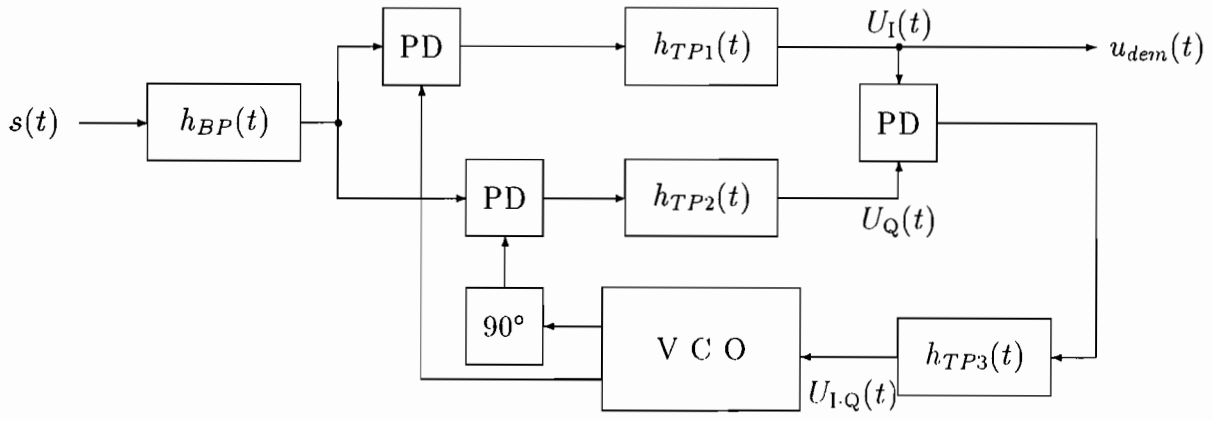


Bild 4.4.2 Costas Regelschleife mit (obligatorischem) Eingangsbandpaß ($h_{BP}(t)$), bestehend aus: drei Phasendetektoren (PD), drei Tiefpässen ($h_{TP}(t)$) und einem spannungsgesteuerten Oszillator (VCO) mit 90° Phasenschieber.

Als weiterer Vorteil der kohärenten Empfänger können die im allgemeinen um ca. 1 dB bessere S/N–Werte bei gleicher Bitfehlerwahrscheinlichkeit [Lü88, S. 181] genannt werden. Dies setzt allerdings einen ständig eingerasteten Regelkreis voraus, was vor allem bei schlechten S/N–Werten nicht gegeben ist. Dies kann bei einem starken Störsender in der Nähe des Nutzsignalfrequenz dazu führen, daß der Empfänger auf das Störsignal einrastet und die Demodulation unbrauchbar wird [Hil93, S. 118-124], [Wie88, S. 50].

Da in der Literatur teilweise Begriffe wie Halte– und Fangbereich etwas unterschiedlich gefaßt sind, soll im folgenden kurz die verwendete Terminologie nach [Gar79] anhand der linearen PLL beschrieben werden. Die Aufwandsbetrachtungen und eine Beschreibung des dynamischen Verhaltens der realisierten Empfängerschaltungen beenden diesen Abschnitt.

4.4.1 Linearer Phasenregelkreis

Lineare und digitale Regelschleifen unterscheiden sich insbesondere durch die Wahl des Phasendetektors (PD). Bei der **linearen** PLL/CL wird als PD ein Multiplizierer eingesetzt; das Eingangssignal ist dabei mehrstufig oder analog — mit **digitalen** PLL/CL können nur binäre Eingangssignale verarbeitet werden. Die Bezeichnung „digital“ bezieht sich demnach nicht auf die Realisierungsform, sondern auf die Beschaffenheit der Eingangssignale.
Eine lineare PLL besteht aus den Hauptblöcken:

- Multiplizierer als Phasendetektor
- Schleifenfilter
- spannungsgesteuerter Oszillator (VCO)

Für ein Einrasten des Regelkreises ist es notwendig, daß ein S/N $\geq 4 = 6$ dB am Ausgang vorhanden ist [Bes87, S. 35]. Da im allgemeinen die Selektion im Antennenkreis und im Schleifenfilter der PLL dafür nicht ausreichend ist, ist es zwingend notwendig, wie die Untersuchungen in [GS93] bestätigen, eingangsseitig eine zusätzliche schmalbandige Selektion vorzusehen. Zur Verdeutlichung wurde deshalb, obwohl der BP nicht elementarer Bestandteil von PLL/CL ist, ein BP in die Abbildungen 4.4.1 und 4.4.2 mit aufgenommen. Die Filterung wird wesentlich vereinfacht, wenn man ein feste Zwischenfrequenz wählt [Bra90]. Setzt man digital steuerbare Mischfrequenzerzeugung und S/C–Filter ein, so ist es möglich, sehr flexibel eine optimale Anpassung an die jeweiligen Empfangsverhältnisse vorzunehmen [Sch91a]. Bei einer vollständig digitalen Realisierung der linearen PLL spricht man dann von einem „Digital Controlled Oscillator“ (DCO) statt von VCO.

Nach diesen grundsätzlichen Betrachtungen sollen im folgenden die wichtigsten Kenngrößen des dynamischen Verhaltens der Regelschleifen beschrieben werden.

Der VCO/DCO schwingt mit der Kreisfrequenz $\omega_2 = \omega_0 + K_0 \cdot U_f(t)$, wobei ω_0 die Ruhefrequenz und K_0 dessen Verstärkungsfaktor bezeichnet. Liegen sinusförmige Signale vor, so entsteht als Tiefpaß–Ausgangssignal

$$U_d(t) = U_{Tiefpaß} = K_d \sin\{\Delta\phi(t)\} \tag{4.4.1}$$

wobei $\Delta\phi(t)$ die Phasendifferenz zwischen $U_1(t)$ und $U_2(t)$ beschreibt. Für kleine Differenzen („In Lock") kann der Sinus durch sein Argument angenähert werden und $U_d(t)$ ist proportional $\Delta\phi(t)$.

Nun ist es leider nicht immer so, daß der Phasenregelkreis bei einem Phasen- oder Frequenzsprung des Eingangssignals in der Lage ist, diesen vollständig auszuregeln. Solange das System stabil ist, also der VCO dem Eingangssignal genügend schnell folgen kann, bezeichnet man die PLL/CL als „eingerastet" oder „synchronisiert". Bei zu großen oder zu schnellen Änderungen der Frequenz des Eingangssignals kann die Regelschleife jedoch „außer Tritt" geraten. Man sagt in diesem Fall, die Regelschleife ist „ausgerastet" („out of lock").

In Abbildung 4.4.3 sind die Betriebsbereiche dargestellt, in denen sich die PLL — je nach Differenz $\omega_1 - \omega_2$ — befinden kann. Der **Haltebereich** $\omega_0 \pm \Delta\omega_H$ („hold in range"), auch statische Stabilitätsgrenze genannt, ist derjenige Bereich, innerhalb dessen die PLL bei sehr langsamem „Wobbeln" der Eingangsfrequenz ein stabiles Arbeiten der PLL auftritt; meßtechnische Bestimmung siehe z. B. [Ges84, S. 13]. Geht man jedoch von einer nicht eingesrasteten PLL aus, so beschreibt der **Fangbereich** $\omega_0 \pm \Delta\omega_L$ („Lock–in range") den Betriebsbereich, bei dem die PLL innerhalb einer Schwebung $\omega_1 - \omega_2$ einrastet; meßtechnische Bestimmung siehe z. B. [Bes87, S. 26].

Im sogenannten **Ziehbereich** $\omega_0 \pm \Delta\omega_L$ („Pull–in range") der PLL rastet die PLL nach der Fangzeit T_L auf das Eingangssignal ein. Diese Fangzeit kann mehrere Schwebungsperioden $\omega_1 - \omega_2$ umfassen. Außerdem wird noch der **Ausrastbereich** $\omega_0 \pm \Delta\omega_{PO}$ („Pull–out range") als maximaler Frequenzsprung spezifiziert, der noch zu keinem Ausrasten der PLL führt und deshalb auch die **dynamische Stabilitätsgrenze** bildet.

In zahlreichen Veröffentlichungen werden diese Betriebsbereiche für verschiedene Typen von PD, Schleifenfilter und VCO berechnet und optimiert, siehe z. B. [LC81], [SR64], [Ste88], [MT82] oder [MT81].

Der Vorteil der *linearen* Regelkreises gegenüber der *digitalen* PLL/CL liegt in der Störbefreiung [Bes87, S. 34]. Gegenüber der DPLL ergibt sich allerdings ein größerer Hardware–Aufwand insbesondere wegen des als Multiplizierer auszuführenden Phasendetektors. Außerdem zeigt dieser PD eine labile Gleichgewichtslage bei 90° (stabil −90°), was das Einrastverhalten der PLL erschwert [Ges84, S. 8]. Benutzt man die Funkti-

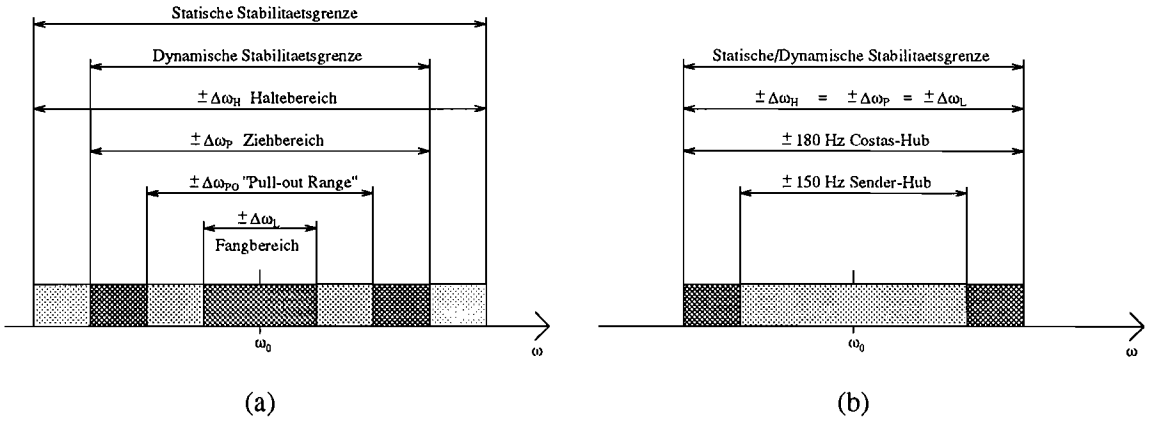


Bild 4.4.3 (a) Mögliche Betriebsbereiche einer PLL/CL (b) Tatsächlich realisierte Betriebsbereiche der CL entsprechend Bild 4.4.6.

onsblöcke entsprechend [AW91] für die Realisierung einer linearen PLL, so ergeben sich für 8 Bit breite Eingangswerte die Abschätzungen entsprechend Tabelle 4.4.1 für den Realisierungsaufwand in CLBs. Dabei ist in der rechten Spalte berücksichtigt, daß für einen universellen AM–, PM– und FM–Demodulator eine zweiter CIC–TP und ein zusätzlicher Multiplizierer benötigt wird, der durch Multiplexbetrieb des Array–Multiplizierers realisiert ist.

Funktionsgruppe	Nur FM	FM, AM und PM
Phasendetektor (Multiplizierer $8 \cdot 8$ Bit)	65	72
Schleifenfilter (2 stufiges CIC)	84	168
Abtast–Frequenzerzeugung	34	34
DCO ($N/N + K$–Teiler, sin/cos–Tabelle)	16	16+2
Ablaufsteuerung und TMS–Interface	15	15
Summe	214	307

Tabelle 4.4.1 Auswandsabschätzung für die Realisierung eines linearen universellen PLL–Demodulators mit Xilinx FPGAs.

Bei der Aufwandsbetrachtung im Vergleich zu dem inkohärenten Null–ZF–Empfänger (Bild 4.1.2 bzw. Tabelle 4.1.1) ist zu berücksichtigen, daß mit dem hier vorgeschlagenen PLL–Empfänger *keine* weiteren Demodulationschritte in einem nachgeschalteten DSP ausgeführt werden müssen, sondern direkt die AM–, PM– bzw. FM–Demodulationswerte vorliegen. Für reine PM/FM–Signale kann durch eine digitale PLL eine weitere deutliche Aufwandsreduktion erreicht werden.

4.4.2 Die digitale PLL

Wie bereits im Abschnitt 4.4.1 erläutert wurde, spricht man von einer digitalen PLL, wenn die Phasendetektoren nur binäre Signale verarbeiten. Von den **Phasendetektoren** sind EXOR–Gatter, flankengesteuertes JK–Flip–Flop und ein Detektor aus einigen Gattern und zwei RS–Flip–Flops bei PLL–ICs gebräuchlich [Bes87, PLL–IC–Übersicht, S. 60-65]. Der Phasendetektor aus Abbildung 4.4.4 ist zwar der aufwendigste PD, jedoch ist nur dieser PD frequenz– und phasensensitiv und besitzt zudem einen unendlich großen Halte– und Fangbereich.

Als volldigitale **Schleifenfilter** bei DPLLs kommen zumeist modifizierbare Teiler zum Einsatz. Dies können einzelne $N/(N + K)$–Teiler oder auch Mehrfachanordnungen, wie z. B. der „N vor M"–Teiler sein, bei dem je ein Zähler für UP und DOWN eingesetzt wird und ein zusätzlicher Zähler für die Summe der UP/DOWN–Impulse. Es wird so bewirkt, daß Steuerimpulse erst dann an den DCO weitergegeben werden, wenn der Phasenfehler einen bestimmten Betrag überschritten hat.

Als **digitale steuerbare Oszillatoren** (DCO) kommen praktisch alle Frequenzgeneratoren aus Kapitel 2 in Betracht. Einschränkend sind für die Auswahl der DCOs bei LW–Demodulation folgende Gesichtspunkte:

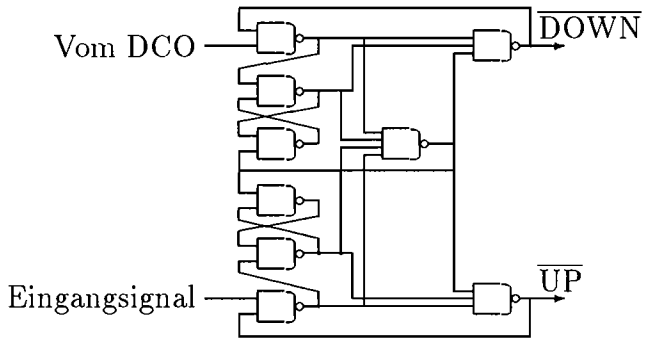


Bild 4.4.4 Phasendetektor nach [Xil93, S. 8-127].

- geringer Frequenzhub des Modulationssignals/kleiner Modulationsindex, siehe auch Bild 4.3.3
- kurze Einschwingzeit ($T_L < 0,5$ ms)
- wenige Trägerperioden pro Regelintervall
- hohe Frequenzauflösung (ca. $300/15 = 20$ Hz bei Graustufenbildern)

wobei der Frequenz– bzw. Phasenfehler noch wesentlich unterhalb 20 Hz liegen sollte.

Die als DCO meist verwendete Variante stellt der veränderliche Teiler dar, da er ein sehr geringes Phasenrauschen besitzt. Die oft ungenügende Frequenzauflösung des Teilers als DCO wird durch eine niedrig liegende ZF des Empfängers verbessert [RS93].

Die DPLLs mit dem geringsten Aufwand an CLBs arbeiten nach dem Verfahren der „Pulse Steal" Technik. Eine solche DPLL wird im hochintegrieten Baustein 74LS297 eingesetzt [TG]. Hierbei liegt der EXOR–Phasendetektorausgang am Eingang des $\overline{UP}/DOWN$–Zählers, der das Schleifenfilter darstellt. Der Carry– bzw. Borrow–Ausgang dieses Zählers bewirkt ein Generieren bzw. Löschen von Impulsen eines I/D–Zählers. Der zweite Teil des DCO besteht aus einem festen $\div N$ Teiler, der die Regelschleife zum PD hin schließt. Das „Stehlen" von Impulsen des DCOs ist der Grund für die Bezeichnung diese PLL–Technik: „Pulse Steal" Technik.

Die Schaltungsidee des Bausteins 74LS297 läßt sich noch verbessern bzw. vereinfachen. Zum einen wird der EXOR–PD durch ein phasen– *und* frequenzsensitives JK–FF ersetzt, und die Teileranordnung kann weiter vereinfacht werden. Man gelangt dann zu der Realisierung nach [McC92], siehe Bild 4.4.5. Die Schaltung arbeitet folgendermaßen: Das Detektor–Flip–Flop „Detect" arbeitet bei einer Ruhefrequenz

$$F_{comp} = \frac{F_{in}}{N} = \frac{F_{osc}}{KM} \tag{4.4.2}$$

Um auch bei höher liegenden Eingangsfrequenzen einrasten zu können, wird die Oszillatorfrequenz F_{osc+} etwas höher als die Ruhefrequenz gewählt

$$T_{comp} - T_{comp+} = \frac{1}{2} T_{osc}, \tag{4.4.3}$$

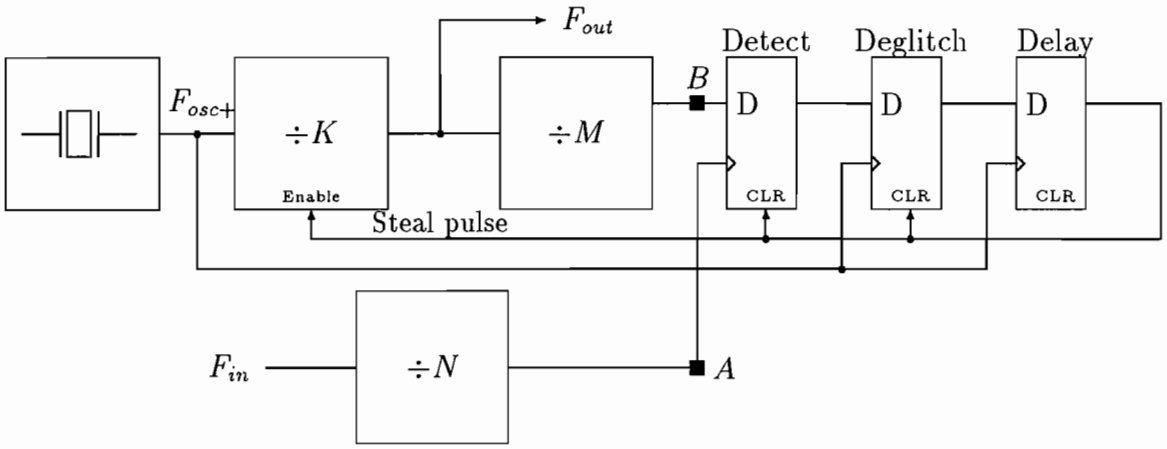


Bild 4.4.5 PLL in „Pulse Steal" Technik [McC92].

so daß das Signal am Punkt B um eine halbe Periode von F_{osc} weiter voreilt und bei der Ruhefrequenz nach rund 2 Taktperioden eine „1" in das Detektor–FF eingeschrieben wird. Diese durchläuft daraufhin die nachgeschaltete Flip–Flop–Kette und sorgt für das Auslöschen eines Zählimpulses im Teiler K („Pulse stealing"), so daß das Signal am Punkt B wieder derart verzögert wird, daß das Signal am Punkt A nacheilt und der ganze Zyklus sich wiederholt und die PLL eingerastet bleibt. Die Ausgangsfrequenz läßt sich auf einfache Weise durch die beiden Teiler K und M variieren, was aber bei Verwendung der PLL als Demodulator unerheblich ist. Der *Fangbereich* dieser PLL ist nach unten quasi unbegrenzt $F_{in}|_{min}$=0 Hz. Nach oben ist er durch die maximale Ausgangsfrequenz F_{osc+}/K begrenzt. Dies ergibt einen Fangbereich bezüglich des Eingangssignal von

$$\pm\Delta\omega_L \;=\; \pm N \cdot F_{osc+}/(K \cdot M) \tag{4.4.4}$$

In [RS93] wurde eine DPLL in vereinfachter „Pulse Steal" Technik eingesetzt. Dabei wurde der Teiler N und M nicht realisiert und die zweite Zwischenfrequenz des Doppelsuperhet–Empfängers so gewählt, daß dem Frequenzhub von 300 Hz genau 32 Stehlimpulse entsprechen. So konnte die sich anschließende digitale Umsetzung in Graustufen stark vereinfacht werden. Bei der realisierten Pixel–Rate von 1920 Baud und der gewählten ZF von 16,6 kHz werden jeweils vier „Stehlwerte" pro Bildpunkt ausgewertet, so daß pro Messung die addierten Stehlwerte letztlich um ld(2 · 4)=3 Stellen binär skaliert (entspricht Division durch 8) werden müssen. Der Tabelle 4.4.2 kann man entnehmen, daß die realisierte PLL relativ effizient in einem preisgünstigen XC3042–FPGA realisiert werden konnte. Die Empfangsqualität erreicht durchaus die der besten inkohärenten Verfahren, siehe Abschnitt 4.5.

4.4.3 Costas Regelschleife

Diese erweiterte Form der Phasenregelschleife geht auf den Amerikaner John P. Costas zurück, der im Jahre 1956 diese Regelschleife erstmalig für die Trägerrückgewin-

Funktionsgruppe	CLB–Aufwand
DCO	16
Phasendetektor	5
Schleifenfilter	6
Mittelwertbildung	11
PC–Interface	10
Frequenzsynthese	26
Ablaufsteuerung	10
Summe	84

Tabelle 4.4.2 Aufwandbetrachtung einer in „Pulse Steal" Technik realisierten DPLL [RS93].

nung [Cos56], [MG92, S. O25] einsetzte. Aus dem Blockschaltbild 4.4.2 entnimmt man, daß eine Costas–Regelschleife im Prinzip aus zwei in Quadratur zueinander arbeitenden PLL–Zweigen aufgebaut ist. Durch den zusätzlichen 90°–Phasenschieber sowie den dritten Phasendetektor und Tiefpaß ergibt sich ein nahezu doppelt so großer Schaltungsaufwand im Vergleich zur herkömmlichen PLL. Dafür bietet die CL, wie im folgenden gezeigt wird, eine Verdoppelung der Einrastgeschwindigkeit gegenüber der PLL.

Zum Verständnis soll die Funktionsweise der linearen Costas Loop anhand von sinusförmigen Eingangssignalen der Form $U(t) = A(t)\sin(\omega_0 t + \Delta\phi(t))$ veranschaulicht werden. Allerdings zeigt sich hierbei, daß die CL besonders empfindlich auf nicht vollständig identische I/Q–Zweige reagiert. Deshalb sollte in *jedem* Fall die CL *volldigital* realisiert werden [Hol82], [Cho87].

Die Tiefpaßsignale $U_{\mathrm{I}}(t)$ und $U_{\mathrm{Q}}(t)$ entstehen durch Mischen des Eingangssignals mit dem sinusförmigen Oszillatorsignal $\sin(\omega_0 t)$ bzw. dem um 90° verschobenen Oszillatorsignal $\cos(\omega_0 t)$ und anschließender Tiefpaßbildung. Für die Tiefpaßsignale ergibt sich:

$$U_I(t) = K_d A(t)\cos(\Delta\phi(t)) \tag{4.4.5}$$

$$U_Q(t) = K_d A(t)\sin(\Delta\phi(t)) \tag{4.4.6}$$

wobei $2K_d$ die Verstärkung der Phasendetektoren darstellt. Nun werden noch $U_{\mathrm{I}}(t)$ und $U_{\mathrm{Q}}(t)$ im dritten Phasendetektor miteinander multipliziert und die TP–Filterung schließt die Regelschleife zum DCO. Als Ansteuerung für den DCO ergibt sich:

$$U_{I\cdot Q}(t) \sim K_d \sin(2\Delta\phi(t)) \tag{4.4.7}$$

Es wird deutlich, daß bei vorhandener AM–Modulation zwingend notwendig ist, daß der Phasendetektor $\mathrm{PD}_{I\cdot Q}$ in der Übersteuerung arbeitet, während der Phasendetektor PD_{I} „linear" arbeiten muß. Liegen reine PM– oder FM–Signale vor, so können alle Phasendetektoren als digitale† Phasendetektoren ausgeführt werden. Eine konkrete Realisie-

† „Digital" bedeutet in diesem Zusammenhang wie bei der PLL, daß der PD nur binäre Eingangssignale verarbeiten kann.

rung einer digitalen Costas Regelschleife ist z. B. in [Bla] beschrieben, wo eine digitale CL für die Demodulation von RDS–Signalen (1 kBaud BPSK–Signale bei einer Mittenfrequenz von 57 kHz im UKW–FM–Basisband) eingesetzt wird.

Der Vergleich von Gleichung (4.4.1) mit Gleichung (4.4.7) zeigt, daß für kleine Werte von $\Delta\phi(t)$ die Steilheit des Einrastsignals $U_{I,Q}(t)$ im eingerasteten Zustand für eine Costas–Loop doppelt so groß ist wie für die PLL.

Es sei noch angemerkt, daß hierbei eine Aufspaltung in I– und Q–Phase erfolgt, wie es vom inkohärenten Empfänger bekannt ist. Da eine I– und Q–Phase bereits vorliegt, kann also die CL direkt als universeller Demodulator für AM–, FM– und PM–Signale eingesetzt werden. Andererseits zeigt diese I/Q–Darstellung deutlich, daß es ebenso möglich ist, statt dem Multiplizierer mit Tiefpaß eine trägerkohärente Hilbert–Abtastung und CORDIC–Wandlung, wie sie im Abschnitt 4.1.4 vorgestellt wurde, durchzuführen, um so den Phasenverlauf des Empfangssignals zu bestimmen und gegebenenfalls die Abtastzeitpunkte „nachzuregeln". Allerdings scheint diese Art des Phasendetektors der Zielsetzung, einen Demodulator mit geringem Aufwand zu realisieren, gegensätzlich zu sein, und soll deshalb an dieser Stelle nicht weiter untersucht werden. Für kompliziertere Signalkonstellationen, wie z. B. 16QAM, die allerdings z. Z. noch nicht bei der LW–Ausstrahlung eingesetzt werden, ergibt die CORDIC–PD Variante durchaus einen akzeptablen Aufwand [ACM82, S. 361], [MG92, S. Q61].

Bei einer volldigitalen Lösung sind prinzipiell für PD, Schleifenfilter und DCO alle Varianten, die im letzten Abschnitt für PLLs beschrieben wurden, möglich. Die folgenden Betrachtungen sind auf die in [GS93] realisierte volldigitale CL beschränkt. Abbildung 4.4.6 zeigt das Blockschaltbild der realisierten Schaltung. Das Eingangssignal ist, wie in Bild 4.4.2 veranschaulicht, ein mit einem analogen Butterworth–Bandpaß vierter Ord-

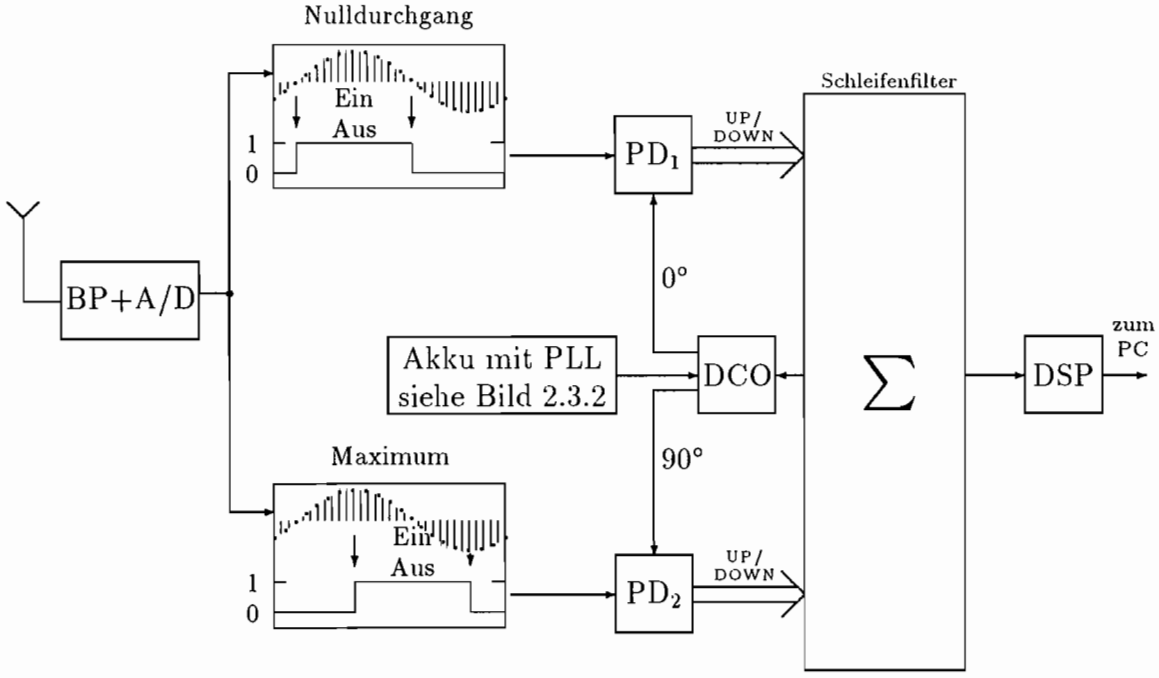


Bild 4.4.6 Struktur der Costas–Loop.

Akkumulator					
Unterlauf	Überlauf	Summe	DCO–IN	Graustufe	$f_{\text{Träger}} \pm \delta f$
Ja	Nein	$s < -(2^{13}-1)$	3	0	+180 Hz
Nein	Nein	$-(2^{13}-1) \leq s < -2048$	2	0	+120 Hz
Nein	Nein	$-2048 \leq s < -512$	1	4	+60 Hz
Nein	Nein	$-512 \leq s < 512$	0	8	+0 Hz
Nein	Nein	$512 \leq s < 2048$	-1	12	-60 Hz
Nein	Nein	$2048 \leq s < 2^{13}-1$	-2	15	-120 Hz
Nein	Ja	$s \geq 2^{13}-1$	-3	15	-180 Hz

Tabelle 4.4.3 Ausgangssignale des Schleifenfilter und Stellinformation des DCO bei 32–facher Überabtastung von DCF37 oder DCF54.

nung (3 dB Bandbreite ca. 2 kHz) gefiltertes und verstärktes LW–Antennensignal. Dieses Signal wird in einem 8 Bit Wandler bei 32– oder 64–facher Nyquist–Frequenz digitalisiert. Das Digitalsignal wird aufgespalten und einem Nullstellendurchgangsdetektor bzw. einer Extremwerterkennung zugeführt, siehe Bild 4.4.6. Die Phasendetektoren vergleichen nun diese Signale mit den Signalen DCO–0° und DCO–90°. Dabei bestehen die Phasendetektoren aus jeweils 2 Flankendemodulatoren entsprechend Bild 4.4.4, so daß insgesamt 4 UP und 4 DOWN–Signale erzeugt werden. Die Phasendetektoren erzeugen dabei UP–Informationen, wenn das Referenzsignal eine zu kleine Frequenz aufweist und DOWN–Informationen, wenn das Referenzsignal mit einer zu großen Frequenz schwingt. Die Differenz der UP/DOWN–Informationen wird im Schleifenfilter über die Dauer eines Pixels in einem 13 Bit Akkumulator ($\hat{=}$ Schleifenfilter) aufsummiert. Diesem Schleifenfilter ist ein Pixel–Converter nachgeschaltet, der die Stellinformation für den DCO liefert, siehe Tabelle 4.4.3. Diese Stellinformation wird ebenso an den DSP übergeben, da dieser Wert unmittelbar dem FM–Demodulationswert entspricht. Der DSP führt dann bei 64–facher Überabtastung noch eine Mittelung über zwei Werte durch und schickt diese Grauwerte seriell an den PC, der zur graphischen Darstellung der WeFAX–Bilder dient.

Aufgrund der festen Überabtastung ergibt sich eine begrenzte Frequenzauflösung dieser CL–Realisierung. Die detektierbare Phasenverschiebung ist bei 32–facher Überabtastung für DCF37–Signale $t_{ph37} = 1/(32 \cdot 117 \text{ kHz}) = 266$ ns. Für die am nächsten zur Trägerfrequenz liegende Frequenz ergibt sich bei 2 kBaud Pixel–Rate

$$f_{\text{Träger+1}} = \frac{1}{1/f_{\text{Träger}} - t_{ph37} \cdot 2\ kBaud/f_{\text{Träger}}} = 117,46 \text{ kHz} \qquad (4.4.8)$$

und es ergibt sich eine Frequenzauflösung von (117,46-117,4) kHz = 60 Hz. Bei einem Frequenzhub von 300 Hz, siehe Tabelle 1.2.1, sind somit 5 Graustufen unterscheidbar. Entsprechende Rechnungen für das DCF54–Signal ergaben ebenfalls 5 Graustufen [GS93, S. 42-43]. Aufgrund des eingeschränkten Arbeitsbereichs (±3 Abtastwerte) rastet die CL nur im Bereich des Modulationshubs $f_0 \pm 180$ Hz ein, obwohl der PD eigentlich einen

	CLBs bei Abtastrate	
Funktionsgruppe	32	64
Frequenzerzeugung	33	36
Nullstellenerkennung	42	42
Extremwerterkennung	16	16
4 Phasendektoren	8	8
Schleifenfilter	51	51
DCO	12	15
TMS–Interface	11	11
Summe	173	179

Tabelle 4.4.4 Aufwandsbetrachtung zur volldigitalen Costas Regelschleife [GS93, S. 60].

unendlich großen Fang– und Haltebereich besitzt. Bei 32–facher Überabtastung können 5 Graustufen und bei 64–facher Überabtastung 10 Graustufen aufgelöst werden. Eine weitere Geschwindigkeitssteigerung war mit dem eingesetzten XC3164-4ns FPGA nicht möglich. Um eine 128– oder 256–fache Überabtastung zu realisieren, müßte man auf die XC4000 Bausteine ausweichen, da diese mittels „Fast Carry“ die Realisierung des Akkumulators (Schleifenfilter) bis 50 MHz erlauben würden. Außerdem müßte ein schnellerer A/D–Wandler eingesetzt werden.

Eine weitere wichtige charakteristische Größe ist die in Abschnitt 4.4.1 beschriebene maximale Einrastzeit T_L. Für diese Realisierung der CL ergibt sich im ungünstigsten Fall eine Phasenverschiebung von 180° ($\hat{=}$ 16 Abtastperioden bei 32–facher Überabtastung), so daß die CL spätestens nach $\lceil 16/3 \rceil = 6$ Abtasttakten einrastet. Dies entspricht einer Einrastzeit von $T_L \approx 1,5\mu s$. Für 64–fache Überabtastung ergibt sich bei Verdoppelung der maximalen Phasenverschiebung auf 32 Abtasttakte und Verdoppelung der Stellimpulse 2·3 ebenfalls $T_L \approx 1,5\mu s$.

Abschließend zeigt Tabelle 4.4.4 den Realisierungsaufwand dieser Costas Regelschleife für die 32– und 64–fache Überabtastung in CLBs.
Es bleibt festzuhalten, daß die kohärente Costas–Regelschleife bei erhöhtem Aufwand ein besseres Fangverhalten zeigt als die PLL.

4.5 Vergleich der realisierten Empfängerschaltungen

Im letzten Abschnitt dieses Kapitels sollen die realisierten Empfängerkonzepte miteinander verglichen werden. Es erfolgt eine Kurzbeschreibung der Intention der mit Hilfe von jeweils 1-5 Studien/Diplomarbeitern realisierten 9 Empfängerkonzepte. Es wurden dabei (mit einer Ausnahme) Universalempfänger realisiert, die mindestens *zwei* Demodulations-

arten oder Signalfrequenzen verarbeiten können. Die Mehrzahl dieser Empfänger wurde bereits während der Poster–Ausstellung der Funkuhrtagung [Hil93] in Braunschweig vorgestellt. Zunächst soll aber, sofern dies noch nicht in den vorangegangenen Kapiteln bzw. Abschnitten geschehen ist, die in den Tabellen 4.5.1 und 4.5.2 verwendeten wesentlichen Unterscheidungskriterien kurz erläutert werden.

Der **Antennenkreis** kann entweder auf die *magnetische* oder *elektrische* Feldkomponente der Langwelle sensitiv sein. Bei der Realisierung von elektrischen LW–Antennen ergibt sich aber anders als bei den magnetischen Antennen *nicht* die Möglichkeit einer Resonanzanpassung. Wie in [Lin74] gezeigt, sollte die Eingangsstufe kapazitiv hochohmig arbeiten, damit die prinzipiell stärkeren und rauschärmeren Signale der elektrischen Antenne nicht degradieren. Mit einer elektrischen Antenne ist ein **Rundumempfang**, d. h. die Antenne besitzt keine besondere Vorzugsrichtung, möglich. Da die Antenne breitbandigen Empfang ermöglicht, ist ohne zusätzliche Maßnahmen eine **Universalantenne** vorhanden [Ber85, S. 10], [Hil88, S. 163-178] und [Wie88, S. 119-135]. Magnetische Antennen können und sollten bei LW–Empfang in Resonanz betrieben werden. Aufgrund von Temperaturänderungen und Alterung der Bauteile sind aber Güten Q größer als 50 technisch nicht sinnvoll. Für eine Universalantenne für den Frequenzbereich 50-150 kHz kann entweder die Empfangskreisgüte (durch einen Dämpfungswiderstand) reduziert werden, oder man verwendet Kapazitätsdioden (Varaktor–Dioden) in Verbindung mit einem digital einstellbaren D/A–Wandler, um die Mittenfrequenz des Kreises verschieben zu können, wobei eine höhere Güte und damit bessere Störsignalunterdrückung erreicht wird. Um mit der magnetischen Antenne Rundumempfang zu ermöglichen, sind zwei Stabantennen zu koppeln. Es bieten sich die drei Möglichkeiten der elektrischen, magnetischen oder induktiven Kopplung [Hil88, S. 193-200] an, siehe Bild 4.5.1. Die Untersuchungen in [BB92] zeigten, daß sich als *Universalantenne* lediglich die magnetische Kopplung eig-

(a) (b)

(c)

Bild 4.5.1 Zur Koppelung zweier Ferritstäbe (a) elektrisch (b) magnetisch (c) induktiv.

net. Bei der elektrischen Kopplung bestehen Schwierigkeiten bei der Potentialtrennung für die mit einem D/A–Wandler anzusteuernde Koppelkapazität $C_k = 1/(2Q)$ und bei der induktiven Kopplung müßte die Koppelinduktivität variabel sein, was ebenfalls zu Realisierungsproblemen führt.

Da die Empfangspegel im Bereich 20-2000 km der Sendeanlagen um ca. 80 dB variieren [Hil88, S. 257], ist eine **automatische Verstärkungsregelung** (AGC) vorzusehen. Aufgrund der zu realisierenden großen Zeitkonstanten für das Funkuhrsignal kommen hierbei nahezu ausnahmslos 1 bis 2 multiplizierende D/A–Wandler (MDAC) zum Einsatz. Lediglich beim reinen FM–Empfänger für DCF37/54 ([Bra90], [Tam91] und [RS93]), konnte eine (schnellere) analoge AGC verwendet werden, die im allgemeinen preisgünstiger realisierbar ist.

Bei der **Frequenzerzeugung** kam eines der Verfahren aus Kapitel 2 zum Einsatz. Für Festfrequenzerzeugung (z. B. für eine Umsetzung erste ZF nach zweite ZF) konnten Quarzoszillatoren (mit Teiler) eingesetzt werden. Für *verschiedene*, aber bereits festliegende Frequenzen konnte meist eine Teiler–Kombination für Quarz und PLL, siehe Abschnitt 2.3.1, gefunden werden, welche die geforderten Frequenzen liefert, wobei der Teiler in der PLL–Rückführung, um Stabilität zu garantieren, nicht zu groß (< 1000) werden durfte. Lag das Schwergewicht der Empfängerrealisierung in der Universalität, so wurden der verlustleistungsarme und aufwandsgünstige Akkumulator–Synthesizer als PLL–Referenz eingesetzt. Das neue hybride Frequenzsyntheseverfahren (Synthesizer mit Akkumulator–Rest–Division, siehe Abschnitt 2.3.3) wurde hier nicht eingesetzt, da der Realisierungsaufwand etwas größer ist und die exzellent kurze Einschwingzeit † nicht unbedingt gefordert ist.

Die Wahl von **Zwischenfrequenzlage** und **Hauptselektion** hängen unmittelbar mit dem **Demodulationsprinzip** zusammen. So zeigt sich bei *kohärenten* Demodulatoren, daß bereits vor dem Demodulator die Hauptselektion durchgeführt werden muß, um einen ausreichenden Signal/Rauschabstand für das Einrasten des Phasenregelkreises zu erreichen. Bei höher als die Signalfrequenz liegender ZF↑ entfällt die Spiegelfrequenzproblematik und es können keramische Filter eingesetzt werden, die eine hohe Ordnung besitzen und so eine gute Nachbarkanalselektion ermöglichen. Der geringe Frequenzhub der LW–FM–Signale ist bei hochliegender ZF schwierig zu handhaben [Bra90]. Wird eine zweite, tieferliegende ZF verwendet, so zeigt sich, daß sich wesentliche Vereinfachungen bei der Realisierung der Regelschleife ergeben. Beim *inkohärenten* Empfänger hat man die Wahl, die Hauptselektion vor *oder* hinter den Demodulator zu plazieren. Entscheidet man sich für eine frühe Hauptselektion, so muß, da steuerbare schmalbandige HF–Bandpaßfilter in S/C–Technik oder konventioneller MDAC–Realisierung (siehe Abschnitt 3.1) die geforderten Filterordnung bzw. Nachbarkanalselektion nicht erreichen, mindestens eine ZF vorgesehen werden. Für eine digitale Hauptselektion ist eine HF–Filterung (Antennenkreis oder MDAC–Filter) zur Dämpfung der Aliasing–Frequenzen vorzusehen.

Als **Anzeige** der portablen Empfänger kommen alphanumerische und graphische LCD–Displays zum Einsatz. Im Hinblick auf einen geringen Stromverbrauch werden VGA–Graustufen–Displays ohne Hintergrundbeleuchtung für graphische Darstellungen

† Für einen Omega–Navigationsempfänger ist für die Gewinnung der vollständigen Orts–Information ein schnelles Umschalten zwischen verschiedenen Längstwellen im Bereich 10 kHz nötig, so daß hier das neue hybride Verfahren wesentliche Vorteile besitzen würde.

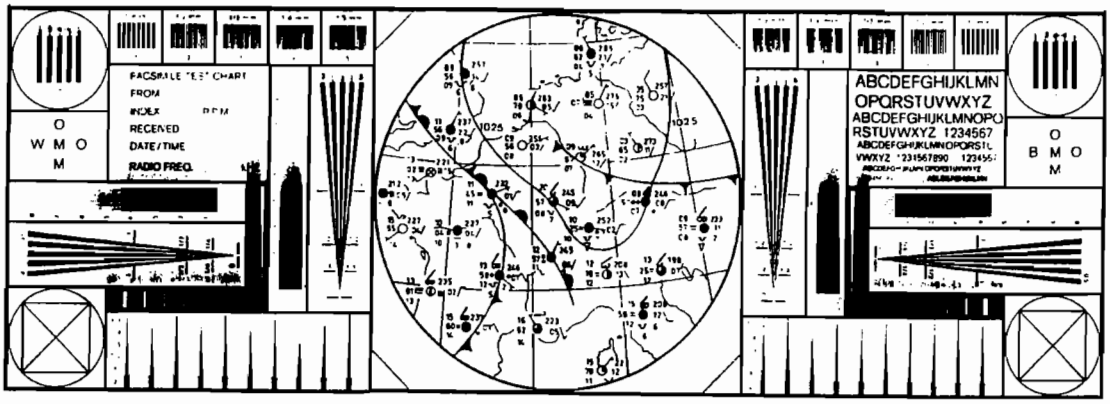


Bild 4.5.2 Um $8^{\underline{13}}$ UTC ausgesendetes Originaltestbild von DCF 37 (117,4 kHz).

verwendet. Für einen nicht portablen Empfänger wurden für verschiedene Rechnersysteme Software–Programme zur Darstellung von Funkuhrdaten und Wetterbildern entwickelt.

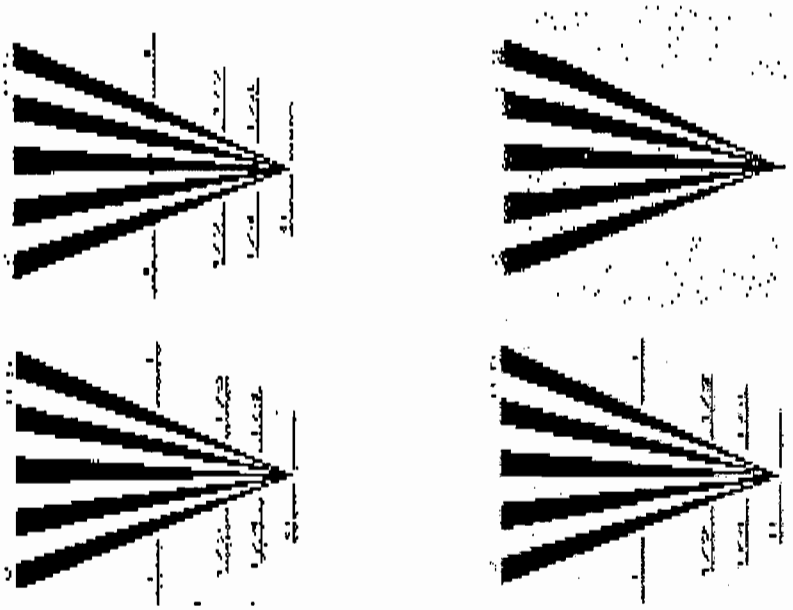
Ein geringer **Leistungsverbrauch** stand insbesondere bei der Entwicklung der portablen Empfänger im Vordergrund. Hier ist, soweit möglich, eine Untergliederung — zwischen analogem Eingangskreis bis hin zum A/D–Wandler auf der einen Seite und Digitalteil auf der anderen Seite — vorgenommen worden.

Als letztes Unterscheidungskriterium ist die erzielte **Bitfehlerwahrscheinlichkeit** (BFW) für Funkuhr bzw. Bildempfang zu nennen. Für das FU–Signal ergab sich mit allen Empfängern für den Testempfang in der Nähe von Kassel (rund 200 km vom Sender entfernt) eine BFW besser 10^{-3}, so daß mit der im nächsten Kapitel beschriebenen Fehlerkorrekturmaßnahmen leicht BFW kleiner 10^{-6} erreicht werden können. Für die Bestimmung der BFW für den Bildempfang wurde in [Hä93] eigens ein Programm erstellt, daß ein empfangenes Bild mit dem zur Ausstrahlung benutzten Originalbild des DWD vergleicht. Dazu wird das Originalbild mit einem „Scanner“ in einen Computer eingelesen. Günstig ist, daß um $8^{\underline{13}}$ UTC vom Sender DCF 37 recht regelmäßig eine Testkarte, siehe Bild 4.5.2, ausgestrahlt wird. Die Bilder 4.5.3(a)-(d) zeigen den gewählten Ausschnitt für den Empfängervergleich der angegeben Empfängertypen.

Leider konnten für die als erstes realisierten zwei Empfänger ([Tam90], [Bra90] bzw. [Sch91a], [Bur91]) keine vergleichenden Messungen mehr durchgeführt werden, da die Bauteile in sich anschließenden Projekten wiederverwendet wurden und die Auswertesoftware zum damaligen Zeitpunkt noch nicht zur Verfügung stand. Eine qualitative Bewertung anhand der vorliegenden empfangenen Wetterbilder ergaben, daß diese Empfänger mit BFW noch unterhalb des CL–Empfängers anzusetzen sind.

Der Vergleich der prozentualen Übereinstimmung mit dem Originalbild ist in den Tabellen 4.5.1 und 4.5.2 angegeben. Dies ist nur als relatives Maß und nicht als BFW anzusehen, da ein wesentlicher Beitrag dadurch entsteht, daß im Übertragungsprotokoll keine Pixelsynchronisation (siehe Abschnitt 1.2.2) vorhanden ist und deshalb insbesondere bei s/w–Übergängen (Kanten) immer ein zusätzlicher Fehler auftritt, obwohl der Empfänger qualitativ hochwertige Bilder liefert, wie z. B. der Typ (5) .

Die Ergebnisse des Bildvergleichs zeigen, daß der hohe Realisierungsaufwand bei volldigitalen Quadraturüberlagerungsempfänger (5) in eine ansprechende Bildqualität umgesetzt wurde. PLL–Empfänger (3) mit guter analoger Selektion und der



(a) (b) (c) (d)

Bild 4.5.3 Ausschnittsvergleich empfangener Bilder in Darmstadt, d. h. rund 30 km vom Sender entfernt. (a) Volldigitaler 0 ZF Quadraturüberlagerungsempfänger (b) Costas–Loop 64–fache Überabtastung. (c) FPGA–PLL. (d) Hilbert–Abtastempfänger 1. Ordnung mit CORDIC–Demodulation durch DSP.

Hilbert–Abtastempfänger mit CORDIC–Demodulation durch DSP (8) erzielen mittlere Empfangsergebnisse. Der direkt umsetzende CL–Empfänger (4) besitzt noch eine unzureichende Analogselektion und erreicht daher nicht die BFW der anderen Empfänger.

Im folgenden werden die verschiedenen Intentionen der durchgeführten Empfängerrealisierungen stichwortartig erläutert. In den Tabellen 4.5.1 und 4.5.2 sind die realisierten Empfänger anhand der genannten Entwurfsmerkmale gegenübergestellt. Detailliertere In-

Nr.	Antenne	Frequenz–erzeugung	ZF	Haupt–selektion	Dem.–Typ	Anzeige	P $\frac{analog}{digital}$	% gleich
(1)	Breitband Ferrit	PLL mit Teiler	1ZF↑	455 kHz Keramisch	PLL s.A.4.4.1	PC	a+d: 0,80 W	< 93
(2)	Ferrit + Varaktor	PLL mit Teiler	1ZF↓	S/C–Filter 2 ×4. Ord.	PLL s.A.4.4.1	PC 2×16 LCD	a:1,185 W d:0,135 W	< 93
(3)	elektr. Langdraht	Akku als PLL–Ref.	1ZF↑ 2ZF↓	keramisch OP–BP	PLL s.A.4.4.2	PC	a:1,32 W d:0,85 W	95,4
(4)	Breitband Ferrit	Akku als PLL–Ref.	0ZF	HF–BP 4. Ord.	CL s.A.4.4.3	PC	a:2,566 W (d:3,65 W)	93,1

Tabelle 4.5.1 Liste der realisierten *kohärenten* Empfängerkonzepte.

formationen sind in den zugehörigen Studien/Diplomarbeiten selbst nachzulesen. In den Projekten (1) bis (4) werden zunächst die kohärenten, dann in (5) bis (9) die inkohärenten Empfänger beschrieben.

(1) Der rein **analoge PLL–Empfänger mit keramischer ZF–Filterung** ([Tam90], [Bra90]) diente als Referenz, welche BFW mit konventionellen einfachen analogen Empfangsprinzipien bei FM–Empfang erreicht werden kann. Ein hoher Abgleichaufwand und mäßige Empfangsqualität bei zugleich kostengünstiger Realisierung charakterisieren diesen Empfänger.

(2) Der **digital steuerbare Analogempfänger mit PLL–Demodulation** ([Sch91a], [Bur91]) ermöglicht eine flexible digitale Einstellung von Bandbreiten, Demodulationsart und Signalfrequenzen. Ein MC steuert den Empfänger und dient gleichzeitig zur Übertragung der AM, FM bzw. PM demodulierten und A/D gewandelten Signale zum PC. Mehrere S/C–Filter und MDACs in Reihe ermöglichen einen großen Dynamikbereich (ca. 100 dB) und sorgen gleichzeitig für gute Aussteuerung und damit geringes Phasenrauschen der S/C–Filterbausteine. Trotz der hohen Universalität war die Empfangsqualität aufgrund von Eigenstörungen durch die digitale PLL–Frequenzerzeugung für die Mischfrequenzen und S/C–Filter nicht so gut wie erwartet. Die Empfangsqualität war nicht zufriedenstellend — könnte durch andere Baugruppenplazierung noch wesentlich verbessert werden.

(3) Bei der Realisierung der **volldigitalen PLL mittels FPGA** ([RS93]) wurde die Vermutung bestätigt, daß auf Kosten eines erhöhten Aufwands in der analogen Filterung (zweite ZF) der Aufwand für den eigentlichen Demodulator geringer ausfällt, siehe auch Abschnitt 4.4.2, als beim inkohärenten Demodulator. Der Einsatz der elektrischen Langdrahtantenne (ca. 10 m) ermöglicht zudem ein etwas verbessertes

Nr.	Antenne	Frequenz–erzeugung	ZF	Haupt–selektion	Dem.–Typ	Anzeige	P $\frac{analog}{digital}$	% gleich
(5)	Breitband Ferrit	Akku als PLL–Ref.	0ZF	dig. 2 Stufen CIC–TP	QÜ s.A.4.1.1	PC	a:2,43 W d:3,65 W	95,7
(6)	Schmalband + Ferrit	Quarz mit Teiler	≈0ZF	analog OP–BP	HA0 s.A.4.1.4	2×16 Stellen LCD	a+d: 0,05 W	—
(7)	2×Ferrit+ 2D/A+Varaktor	Akku mit PLL–Ref.	1ZF↓	HF:MDAC S/C–ZF–BP	HA1 s.A.4.1.4	200×600VGA +2×16 LCD	a:0,35 W d:0,7 W	—
(8)	Ferrit+ Varaktor	PLL mit Teiler	0ZF	dig. IIR 10 Biquads	HA1 s.A.4.1.4	PC	a:0,61 W d:1,09 W	95,4
(9)	Schmalband Ferrit	PLL mit Teiler	0ZF	dig.. FIR 62 Tap	HA1 s.A.4.1.4	280×125 mm LCD	a:0,28 W d:0,87 W	—

Tabelle 4.5.2 Liste der realisierten *inkohärenten* Empfängerkonzepte.

S/N–Verhältnis am Empfängereingang. Die akustische Ausgabe des Signals (zusätzliche zweite ZF bei $\approx$ 2 kHz) über einen Lautsprecher erlaubt zudem eine Veranschaulichung der FM–Modulation — insbesondere können Start– und Stop–Ton gut „gehört“ werden.

(4) Der **volldigitale Costas–Loop–Empfänger** ([GS93]) wurde als Null–ZF–Empfänger realisiert, der somit einen steuerbaren Eingangsbandpaß benötigt. Das im allgemeinen doppelt so schnelle Einrasten der CL gegenüber der PLL bei ungefähr der doppelten Realisierungsaufwand ergab noch einigermaßen zufriedenstellende Empfangsergebnisse.

(5) Der in der Arbeit [AW91] realisierte **volldigitale 0ZF Quadraturüberlagerungsempfänger** ist aufgrund der Anordnung FPGA mit Fließkomma–DSP der flexibelste aller Empfänger. Durch kleine Programmänderungen konnte z. B. von Wetterbildempfang DCF37/54 auf die andere Empfangsfrequenz und Modulationsart des Textsenders DCF49 umgestellt werden. Die Flexibilität des Empfängers zeigt sich außerdem darin, daß die Analysen in Bild 1.2.1(b) und 1.2.3(b) mit diesem Empfänger durchgeführt werden konnten. Der Empfänger erreicht die beste Empfangsqualität aller Projekte. Die Qualität der empfangenen Bilder in Darmstadt (rund 20 km vom Sender) für S/W–Bilder ist so gut, daß diese Bilder unmittelbar auch als Originalbild für den „Bildvergleich“ verwendet werden könnten, siehe Bild 4.5.2.

(6) Der **portable analoge Hilbert–Abtastempfänger nullter Ordnung mit indirekter Zeitauswertung** ([Die91a]) ist der einzige Empfänger, der nur FU–AM–Demodulation durchführen kann. Neben einer kostengünstigen Realisierung der Hilbert–Abtastung war das Hauptziel des Projektes die indirekte Datumauswertung. Sie wird zur Bestimmung von Sonnen/Mond–Auf/Untergangszeiten benutzt. Über Tastatur erfolgt die Eingabe der eigenen Position (Längen und Breitengrad) und die zugehörigen Auf– bzw. Untergangszeiten können mittels Tabelleninterpolation für die nächsten rund 20 Jahre bestimmt werden, wobei die maximale Abweichung ca. 2 Minuten beträgt. Probeempfang ist mit dem eingebauten Akkumulator für ca. 3 Stunden möglich. Die eingebaute Uhrenmodul läuft ohne erneute Akkuladung rund 1/2 Jahr.

(7) Dies umfangreichste aller Projekte des **portablen Hilbert–Abtastempfängers mit CORDIC–Demodulation und LCD–VGA–Display** ([HK92], [BB92] und [Hil92b]) ermöglicht rund 40 Stunden netzunabhängigen Empfang und Anzeige von Funkuhr und/oder Wetterbildern. Die Konzepte für Analog– und Digitalteil wurden besonders in Hinblick auf geringen Stromverbrauch optimiert. Dies reicht von der Auswahl der Operationsverstärker über das Demodulationsprinzip bis hin zur Architektur von CORDIC–Prozessor und VGA–FPGA–Controller. Der 1 MBit Bildspeicher ermöglicht die Darstellung von 500 mal 1000 Bildpunkten bei 4 Graustufen.

(8) und (9) Diese Projekte dienten der Verifikation der in der Patentschrift [MBH92] genannten Vorzüge des **Hilbert–Abtastempfängers mit CORDIC–Demodulation** durch

digitalen Signalprozessor. Hierbei wurden als Alternativen der Hilbert–Abtastempfänger erster Ordnung mit einer mehrfach S&H–Stufe und einem A/D–Wandler (8) im Multiplex [Sch92] und die Alternative mit 3 A/D–Wandlern (9) [Die92] bei gleichzeitiger starker Unterabtastung bezüglich Realisierungsaufwand, Stromverbrauch und Empfangsqualität untersucht, siehe Bild 4.1.15. Dabei wurde der Empfänger [Sch92] zur gedächtnisbehafteten Demodulation von FM–Signalen, siehe Abschnitt 4.3.3, verwendet, während der Empfänger [Die92] für die kombinierte AM/PM–Demodulation des FU–Signals zur Detektion des genauen Sekundenbeginns benutzt wurde. Beide Realisierung konnten auf einer „Europa–Karte“ (100 mal 160 mm) untergebracht werden und lieferten gute Empfangsergebnisse.

Kapitel 5

Codierung zur Verschlüsselung und Fehlerkorrektur

Moderne Übertragungsverfahren, wie Pagingsysteme, Mobilfunk oder Satellitenübertragung, verwenden fast ausnahmslos Algorithmen zur Korrektur von Übertragungsfehlern (Codierung zur Fehlerkorrektur [Dor93]), da es bei bandberenzter Übertragung Codierungsverfahren besser als speziellen Modulationsverfahren gelingt, die Kanalkapazität auszuschöpfen [Lü88, S. 181]. Sicherheitsrelevante Übertragungen werden zumeist mit Verschlüsselungsverfahren versehen, um sie gegen unbefugtes Abhören und eine leichte Veränderung/Falschaussendung zu schützen. Fehlerkorrekturmaßnahmen und kryptographische Algorithmen sind sender– und empfangsseitig zwischen der Datenquelle und der Kanal(de)codierung (Demodulator/Modulator) plaziert, siehe Bild 5.0.1. In diesem Kapi-

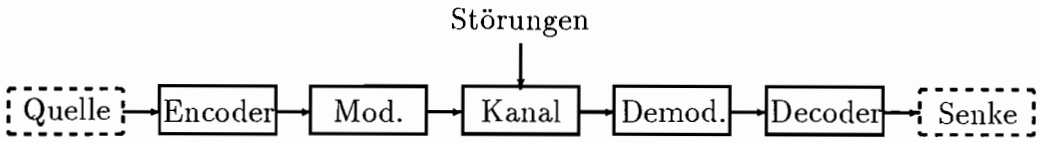


Bild 5.0.1 Übertragungsstrecke mit Fehlerkorrektur und/oder Verschlüsselung.

tel soll untersucht werden, welche Algorithmen zur Fehlerkorrektur und Verschlüsselung besonders effizient mit FPGAs realisiert werden können. Die Untersuchungen werden anhand zweier Anwendungen durchgeführt, die sich mit zukünftigen Übertragungsformaten bei der LW–Datenübertragung befassen. Es werden zugehörige Datenformate entwickelt, günstige Algorithmen bestimmt und Systeme mit Hilfe einer FPGA–Universalplatine in Hardware realisiert [Tam91]. Die zwei Anwendungen sind:

a) Zum bisherigen Übertragungsformat kompatible Fehlerkorrektur für DCF77–Signale

b) Realisierung eines Personenrufsystems mit Datenverschlüsselung und Fehlerkorrektur

Im folgenden werden zunächst die Ergebnisse der Untersuchungen zu **a)** behandelt [Her91a], die bereits teilweise in [GHMB93] veröffentlicht wurden. Danach wird auf die Resultate zu **b)** eingegangen, wo Ergebnisse der Untersuchungen aus [Wat93] und

[MS93] präsentiert werden. An dieser Stelle soll das besondere Augenmerk auf den jeweiligen Aufwand der Algorithmen in CLBs gelegt werden, während die umfangreiche Theorie zu den Algorithmen nur insoweit erörtert wird, daß die verwendeten Begriffe und Formelzeichen deutlich werden. Zu der Theorie existiert eine Reihe guter Lehrbücher, deren Kenntnis an dieser Stelle aus Platzgründen vorausgesetzt wird, wie z. B. [Bla84], [Wig88], [Swe91], [Dor93] oder [FR88].

5.1 Übertragungsprotokoll kompatible, FPGA–Implementierung einer Fehlerkorrektur für DCF77 Signale

Es werden verschiedene formatkompatible Fehlerkorrekturmaßnahmen für das bestehende Minutenprotokoll des DCF77 Senders besprochen. Das sehr robuste Protokollfeldverfahren und ein t–Fehler korrigierender Code werden gegenübergestellt. Es wird sich zeigen, daß bei einer systematischen Codierung von 44 Bits Informationslänge mit zusätzlichen 13 Parity–Bits lediglich 2 Fehler und eine weitere Auslöschung korrigiert werden können. Fehlerschranken und die Konstruktion des (57,44,6)–BCH–Codes werden besprochen. Einfache Schaltungen für Coder und Decoder werden angegeben, und der Realisierungsaufwand verschiedener FPGA–Implementierungen besprochen [Her91a], [dR93]. Da mit der Fehlerkorrektur die Empfangsdauer eines Empfängers bei gleicher Zuverlässigkeit wesentlich verringert wird, ist es nun möglich, den Leistungsverbrauch gegenüber der Protokollfeldauswertung erheblich zu senken. Abschließend werden noch die Kombination von Protokollfeldverfahren und Codierung zur Fehlerkorrektur besprochen.

Grundsätzliche Betrachtungen

Für Entfernungen ab ca. 1000 km ist der Einsatz von Fehlerkorrekturmaßnahmen unabdingbar, da sonst die angezeigte Zeitinformation nicht zuverlässig genug ist. Es wird im folgenden gezeigt, wie das bisherige Übertragungsformat für Korrekturmaßnahmen ausgenutzt werden konnte, welche Vorteile der Einsatz einer zusätzlichen Codierung erwarten läßt und welche Fehlerkorrekturschaltungen für die PGA–Technik besonders geeignet sind.

Das Codierschema für DCF77–Signale wurde bereits in Abschnitt 1.2.1 erläutert und soll hier nicht nochmals im Detail erklärt werden. Für Codierzwecke erscheinen folgende Anmerkungen zum Codierschema sofort einsichtig:

a) Zwei aufeinanderfolgende (fehlerfreie) Minutentelegramme können durch Inkrement/Dekrement der untersten Minutenstelle ineinander überführt werden.

b) Die Bits 28, 35 und 58 sind Parity–Bits, die eine gerade Parität der Bits 21-27, 29-34 bzw. 36-57 bilden mit deren Hilfe eine *Fehlererkennung* möglich ist.

c) Die Bits 0-14 sind im Codierschema des Minutentelegramms nicht belegt[†] und können somit zur *Fehlerkorrektur* eingesetzt werden.

[†] Da in Ausnahmefällen Bit 5 und 6 bei Wartungsarbeiten an der Antenne gesetzt werden ([Bel89, S. 7]), sollten möglichst nur 13 Bit zur Fehlerkorrektur benutzt werden.

Zunächst bleibt die Frage offen, warum nicht schon von Beginn der Übertragung an (1971) eine Fehlerkorrektur vorgesehen wurde, da die Bits 0-14 bislang „ungenutzt“ übertragen werden und der Fehlerschutz somit „nichts gekostet“ hätte.

Im folgenden sollen die Punkte a) bis c) aus Codierungssicht genauer betrachtet werden. Die Codierungstheorie ist hierbei ein leicht und zugleich auch schwer verständliches Themengebiet — leicht, wenn plausible und anschauliche Lösungen betrachtet werden wie: „Durch Hinzufügen von Prüfstellen können Fehler korrigiert werden“ — schwierig dann, wenn die komplexe Algebra der endlichen Felder im Detail betrachtet wird, um Codes zu konstruieren und Eigenschaften von Codes zu belegen.

Wir wollen uns auf die pragmatische Seite der Anwender von existierenden Codes stellen und anschauliche Lösungen angeben — wollen jedoch bei Code–Konstruktionen exakte Formalismen durchführen, zu deren Verständnis die Kenntnis eines der oben genannten zahlreichen guten einführenden Bücher der Codierungstheorie vorausgesetzt wird. Zur Anwendung der angegebenen Codierschemas sind jedoch elementare Kenntnisse der digitalen Schaltungstechnik ausreichend.

5.1.1 Protokollfeldauswertung als Wiederholungscode

Die in Punkt **a)** der Einleitung genannte Vorgehensweise wurde von Hilberg [Hil92a] vorgeschlagen und in [Bel89, Kap. 6.1] ausführlich diskutiert. Das sogenannte Protokollfeldverfahren entspricht der Auswertung eines Wiederholungscodes, da aufeinanderfolgende (fehlerfreie) Minutentelegramme ineinander überführt werden können und daher völlig redundant sind. Für ein Protokollfeld über 5 Minuten ergibt sich die Notation (Information $\Leftrightarrow$ Codewort) für den Wiederholungscode zu

$$0 \Leftrightarrow 00000 \tag{5.1.1}$$

$$1 \Leftrightarrow 11111 \tag{5.1.2}$$

Bei n–facher Wiederholung (Protokoll über n Minuten) entsteht so ein $(n, 1, n)$–Code[‡] mit dem bis zu $\lfloor (n-1)/2 \rfloor$ zufällig auftretende Fehler (Random–Fehler) korrigiert werden können. Das Protokollfeldverfahren ist deshalb sehr robust. Die Fehlerkorrektur ist durch einen einfachen Mehrheitsentscheid durchzuführen.

Von Nachteil ist allerdings, daß der Empfangsteil des FU–Empfängers n Minuten lang „auf Empfang“ sein muß. Da der Leistungsbedarf einer FU nahezu ausschließlich durch die Empfangsdauer bestimmt ist, geht die Batterielebensdauer somit um einen Faktor n zurück. Dies kann bei portablen Systemen zu nicht akzeptablen Leistungsverbrauch/Batterielebensdauern führen, so daß die Protokollfeldauswertung nicht eingesetzt werden kann.

5.1.2 Erweiterung der Fehlererkennung zu einem Hamming–Code

Die vorhandenen Parity–Bits lassen eine Erweiterung zu einem 1–Fehler korrigierenden gekürzten Hamming–Code zu. Die zum Schutz der Minuteninformation zusätzlich

[‡] Die Notation (n,k,d^*) bedeutet: n = Codewortlänge; k = Anzahl der Informationsstellen; d^* = Mindest–Hamming–Distanz, d. h. minimale Differenz zweier beliebiger Codewörter.

benötigten Parity–Bits lassen sich mit der in Bild 5.1.1(a) dargestellten Codier–Schaltung einfach ergänzen.

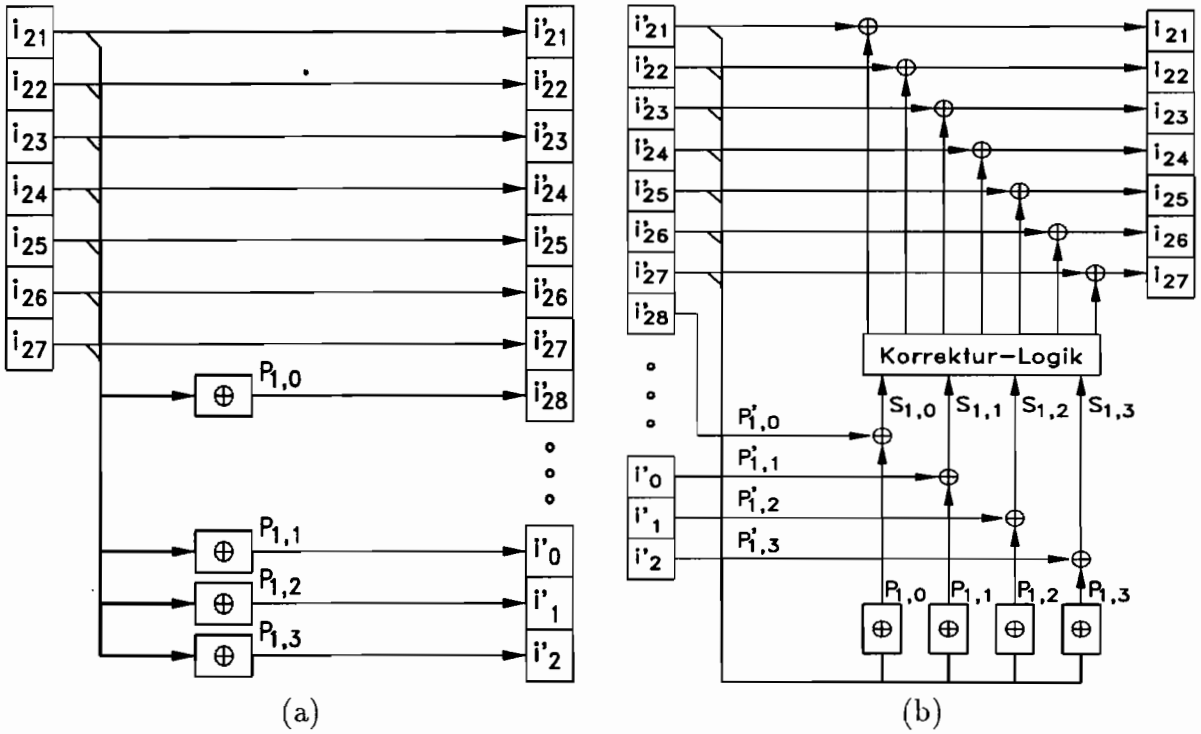


Bild 5.1.1 (a) Coder und (b) Decoder zur Fehlerkorrektur für die Minuteninformation.

Für die Paritäten $P_{1,0}, P_{1,1}, P_{1,2}$ und $P_{1,3}$ sind folgende Additionen modulo 2, d. h. EXOR–Verknüpfung, durchzuführen:

$$
\begin{aligned}
P_{1,0} &= i_{21} \oplus i_{22} \oplus i_{23} \oplus i_{24} \;;\oplus i_{25} \oplus i_{26} \oplus i_{27} \\
P_{1,1} &= i_{21} \oplus i_{23} \oplus i_{25} \oplus i_{27} \\
P_{1,2} &= i_{21} \oplus i_{22} \oplus i_{25} \oplus i_{26} \\
P_{1,3} &= i_{21} \oplus i_{22} \oplus i_{23} \oplus i_{24}
\end{aligned}
$$

Neben dem Parity–Bit $i'_{28}(= P_{1,0})$ sind so weitere drei Paritäten nötig, um mit Hilfe der in Bild 5.1.1(b) angegeben Schaltung die Korrektur für einen Fehler durchführen zu können.

Die im Decoder realisierten Paritäten $(P_{1,0} \cdots P_{1,3})$ sind mit denen im Coder identisch. Nach der Verknüpfung mit den empfangenen Paritäten entsteht das „Syndrom“ $(S_{1,0} \cdots S_{1,3})$. Jedem Syndrommuster ist genau ein Fehlermuster zugeordnet. Tritt z. B. das Muster (1,0,0,1) auf, so ist das Bit 4, d. h. i_{24}, bei der Übertragung gestört und kann korrigiert werden.

Wird die Schaltung in Software in einem MC/DSP realisiert, so bietet sich die Verwendung der Matrixschreibweise an, aus der ebenso die verschiedenen Syndrommuster abgelesen werden können. Für die Parity–Check–Matrix ergibt sich

$$H = \left[P^T : I\right] = \left[\begin{array}{cccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{5.1.3}$$

Nun erkennt man sofort, daß die Spalte des Syndrommusters mit der Position des gestörten Bits identisch ist. Im allgemeinen reicht es aus, die ersten 7 Syndrommuster zu speichern, da es nicht sinnvoll erscheint, die Parity–Bits einer Fehlerkorrektur zu unterwerfen.

Da bei Hamming–Codes prinzipiell die Anordnung der Syndrommuster P (Spalten der H–Matrix) beliebig ist, wurde sie hier so gewählt, daß für die 7 zu korrigierenden Fehlermuster $S_{1,0} = 1$ gilt und so mit Hilfe der Syndromwerte $(S_{1,1} \cdots S_{1,3})$ die „Korrekturlogik“ mit einem einfachen $3 \rightarrow 7$–Decoder realisiert werden kann.

Natürlich kann auch die Codier–Vorschrift in Form einer Generatormatrix formuliert werden. Hier ergibt sich $G = [I{:}P]$, d. h. die für systematische Codierung zwingend notwendige Einheitsmatrix gefolgt von den Verknüpfungen für die 4 notwendigen Paritäten. Ein Codewort v ergibt sich nun aus der Multiplikation des Informationsworts i mit der Generatormatrix G:

$$v = i \cdot G \tag{5.1.4}$$

Die hier für die Minuteninformation durchgeführte Code–Kürzung eines (15,11,3) Hamming–Codes auf einen (11,7,3) gekürzten Hamming–Code kann ebenso für die Stundeninformation durchgeführt werden. Hier entsteht ein (10,6,3)–Code, der einfach durch Streichen der letzten „P^{T}–Spalte“ der $H = [P^{\mathrm{T}}{:}I]$ Matrix konstruiert wird.

Für das Datum (22 Bits) wird ein binärer (31,26,3) Hamming–Code zu einem (27,22,3) 1–Fehler korrigierenden Code gekürzt. Hier kann leider die vorhandene Parität (Bit 58) nicht benutzt werden, so daß 5 zusätzliche Stellen benötigt werden. Es ergibt sich folgende Parity–Check–Matrix:

$$H = \left[\begin{array}{cccccccccccccccccccccccccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Wiederum wurde die Syndromtabelle so sortiert, daß die Korrekturlogik durch einen einfachen $5 \rightarrow 22$–Decoder realisiert werden kann.

In der Tabelle 5.1.1 ist der für diesen ersten Ansatz zur Fehlerkorrektur nötige Realisierungsaufwand angegeben. Zusätzlich müßten für eine konkrete Realisierung noch einige CLBs für das MC/DSP–Interface und eine Ablaufsteuerung berücksichtigt werden.

Insgesamt läßt sich festhalten, daß es durch 3+3+5=11 zusätzliche Parity–Bits bei einem Realisierungsaufwand von ca. 100 CLBs möglich ist, in dem Schema für Minute, Stunde und Datum/Wochentag jeweils ein fehlerhaftes Bit zu korrigieren. Nachteilig bleibt:

- Korrektur von höchstens einem Fehler in den jeweiligen Bereichen (z. B. 22 Bits Datum !) möglich
- Kein Schutz für Bit 15-20 (z. B. Sommerzeit !)

5.1.3 Entwurf eines Fehlerschutzes für das gesamte Minutentelegramm

Theoretische Überlegungen in [Hil88, S. 67] ergaben, daß statt der möglichen ($2^{35} = 3,44 \cdot 10^{10}$) praktisch (nur) $2,1 \cdot 10^8$ verschiedene Minutentelegramme auftreten. Nimmt man die Informationen Zeitzone, Sommerzeit usw. (Bit 15-19) hinzu, so ergeben sich $6,73 \cdot 10^9$ verschiedene Informationsworte. Für den zu entwerfenden Fehlerschutz stehen bis zu 15 Bit d. h. $2^{15} = 3,27 \cdot 10^4$ verschiedene Prüfworte zur Verfügung. Da für einen beliebigen Code mittels Codetabelle Informationsworte und Prüfworte beliebig miteinander kombiniert werden können, stehen somit $3,27 \cdot 10^4 * 6,73 \cdot 10^9 = 2,2 \cdot 10^{14}$ verschiedene Codes zur Verfügung! Aus dieser immensen Anzahl soll nun ein geeigneter Code ausgewählt werden.

Da die Auswahl auf existierende Codes beschränkt werden soll, sei zunächst eine Übersicht von gebräuchlichen Decodern angegeben. Dabei ist leicht einsichtig, daß optimale Techniken mittels „Look–Up–Tables“ bei der nötigen Tabellengröße (im Bereich von mehreren Gigabyte) für einen kompakten Empfänger sofort ausscheiden. Die Abbildung 5.1.2 beschränkt sich daher auf Decodieralgorithmen, die auf kompakte mathematische Berechnungsverfahren zurückgreifen. Der Unterschied zwischen Block– und Baum–Codes ist das speichernde bzw. nicht speichernde Verhalten bei der Codeerzeugung. Beiden Codes wird eine Coderate $R = k/n$ als Quotient aus der Anzahl der Informationsbits zu Gesamt–Bits (Informationsstellen + Prüfstellen) zugeordnet. Zusätzlich ist bei den

	CLB–Aufwand für		
Funktionsblock	Minute	Stunde	Datum
Register	6	6	14
Sydromberechnung	5	5	16
Korrektur–Logik	4	4	22
Ausgangsregister	4	4	11
Summe	19	19	63
Gesamtzahl	101		

Tabelle 5.1.1 Aufwandsbetrachtung zur Korrektur mit Hamming–Code.

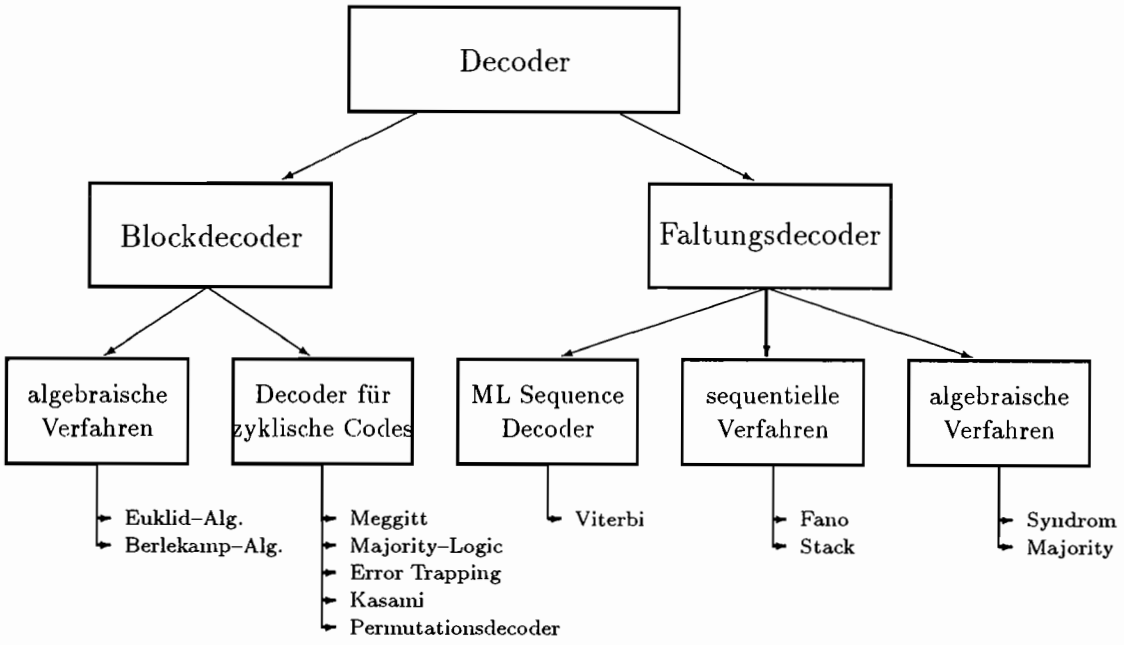


Bild 5.1.2 Übersicht der gebräuchlichen Decoder zur Fehlerkorrektur.

speichernden Baumcodes das n Bit Ausgangswort nicht nur von den k aktuellen Eingangswerten, sondern auch von den m vorherigen Zeichen abhängig. Charakteristisch für Faltungscodes sind die Einflußlänge $\nu = m \cdot k$, sowie das Distanzprofil, die freie Distanz d_f und die minimale Distanz d_m, siehe z. B. [Dor93, S. 142-4]. Während Blockcodes meist mittels der linearen Algebra von Galoisfeldern bestimmt werden können, lassen sich günstige Baumcodes meist nur per Computersimulation bestimmen.

Die Betrachtungen sollen auf **lineare** Codes beschränkt werden, da die Decoder aufgrund der Tatsache, daß bei linearen Codes die Summe zweier Codewörter wieder ein Codewort ergibt, zu wesentlich einfacheren Schaltungen führt. Außerdem kann bei einem linearen Code die Hamming–Distanz allein durch die Distanz zum Nullwort berechnet werden, was eine einfache qualitative Beurteilung erlaubt. Lineare Baumcodes werden auch als **Faltungscodes** bezeichnet, da diese durch Faltung der Informationsbits mit einem FIR–Filter gebildet werden können. **Systematische** Codes liegen dann vor, wenn die Eingangsfolge unverändert in die Ausgangsfolge übernommen wird. Außerdem wird zwischen **katastrophalen** und **nichtkatastrophalen** Faltungscodes unterschieden, wobei sich bei katastrophalen Faltungscodes Falschkorrekturen endlos fortpflanzen können. Es läßt sich zeigen, daß systematische Faltungscodes immer *nichtkatastrophal* sind.

Um die Auswahl weiter einzuschränken, werden, neben der Beschränkung auf die aus der Literatur bekannte linearen Codes, folgende plausible Annahmen getroffen:

a) Der Codierungsaufwand — aber vor allem die Decodierungs–Hardware oder eine äquivalente Software–Realisation — ist mit geringem Aufwand durchzuführen.

b) Es ist ein *systematischer* Blockcode auszuwählen, da einerseits das bisherige Format in den Stellen 15-59 nicht beeinträchtigt werden soll und zum anderen Faltungscodes speziell für diese Anwendung einen zu komplexen Decoder erfordern (siehe **a)**).

c) Bündelfehler (mehrere aufeinanderfolgende Bits sind gestört) sind, wegen der geringen Bitrate und da kein „schnelles Fading" vorliegt, nicht zu erwarten. Deshalb ist ein Code zu wählen, der an zufälligen Positionen im Minutenprotokoll auftretende Fehler korrigiert (Random–Error–Correcting–Code).

d) Keine gute Fehlererkennung, sondern eine möglichst große Anzahl an korrigierbaren Fehlern ist zu wünschen.

e) Es sind minimal 32, aber möglichst 44 Bits zu schützen. Es stehen für die Korrektur 13-15 Bit zur Verfügung.

5.1.3.1 Codierungsschranken

Bevor geeignete Code im Detail beschrieben werden, soll zunächst noch untersucht werden, welche Schranken für die maximal korrigierbare Fehlerzahl bestehen.

Einfache aber zugleich gut zutreffende Schranken sind die „Singleton–Schranke" und die „Hamming–Schranke".

Die **Singleton–Schranke** sagt aus, daß die minimale Hamming–Distanz d^* nach oben durch die Anzahl der Prüfstellen $(n-k)$ begrenzt ist. Allgemein bekannt [Bla84, S. 256] ist auch, daß die Anzahl der zu korrigierenden Fehler t und die Anzahl der korrigierbaren Auslöschungen e für einen Code durch die minimale Hamming–Distanz nach oben beschränkt ist. Dies verbunden mit der Singleton–Schranke führt zu folgender Beschränkung:

$$e + 2t + 1 \leq d^* \leq n - k + 1 \tag{5.1.5}$$

Die Schranke besagt somit, daß bei 15 bzw. 13 Prüfstellen maximal 7 bzw. 6 Fehler korrigierbar sind. Angemerkt sei hier, daß für $d^* = n - k + 1$ von einem **Maximum Distance Separable** Code gesprochen wird. Zugleich sei bemerkt, daß es, abgesehen von dem trivialen Repetition–Code und dem Parity–Check–Code, *keinen* binären „Maximum Distance Separable" Code gibt [Bla84, S. 431].

Für einen t–Fehler korrigierenden binären Code liefert die **Hamming–Schranke** eine enge Beschränkung.

$$2^{n-k} \geq \sum_{m=0}^{t} \binom{n}{m} \tag{5.1.6}$$

Die Gleichung (5.1.6) besagt, daß die Anzahl der verschiedenen Parity–Check–Muster (2^{n-k}) größer oder gleich der Anzahl der verschiedenen Fehlermuster sein muß. Gilt in Gleichung (5.1.6) das Gleichheitszeichen, so spricht man von einem **perfekten Code**. Ein perfekter Code ist z. B. der in Abschnitt 5.1.2 benutzte Hamming–Code. Für die geforderten Randbedingung **e)** ergibt die Hamming–Schranke:

$$n - k = 15; \qquad \begin{cases} k \in 32 \cdots 43 & \rightarrow \quad t = 3 \\ k = \quad 44 & \rightarrow \quad t = 2 \end{cases} \tag{5.1.7}$$

$$n - k = 13; \qquad k \in 32 \cdots 44 \rightarrow t = 2 \tag{5.1.8}$$

Diese einfache Abschätzung zeigt, daß bei diesem Verhältnis von Informations- zu Prüf-Bits 2 bis höchstens 3 Fehler korrigierbar sind. Hier zeigt sich, daß das Minutenprotokoll bezüglich Fehlerkorrektur recht ungünstig gewählt ist. Durch eine binäre Codierung der Darstellung der Zeitdaten und mit Hilfe der Parity-Bits können leicht 8 Bits eingespart werden. Dann könnte ein (59,36,9)-BCH-Code benutzt werden, mit dem 4 Fehler (statt der jetzt möglichen 2 Fehler) korrigiert werden könnten.

5.1.3.2 Auswahl des Codes für das gesamte Minutentelegramm

Wie eingangs erwähnt, sollten möglichst nur 13 Prüfbits zur Fehlerkorrektur benutzt werden, da Bit 5 und 6 ausnahmsweise belegt sein können. Weiterhin sollten möglichst alle 44 Bit geschützt werden. Die Fehlerkorrektur kann bei diesen Randbedingungen maximal 2 Fehler beheben. Da mit den linearen zyklischen binären BCH-Codes eine große Klasse an Codes existiert, für die verschiedene einfache Decodiervarianten im Zeit- und Frequenzbereich aus der Literatur bekannt sind und diese Codes gute Korrektureigenschaften besitzen, wurde ein gekürzter (57,44,6) binärer BCH-Code ausgewählt. Im folgenden wird nun kurz die Konstruktion eines (63,50,6)-BCH-Code mit anschließender Codekürzung zum (57,44,6) Code beschrieben. Der Algorithmus ist in [Bla84, S. 162-6] beschrieben.

Der Code wird durch Transformation von GF(2^6) nach GF(2) entworfen. Eine einfache Darstellung der Codier- und Decodier-Vorschrift gelingt mittels Polynomen. Dazu wird das sogenannte Generatorpolynom benötigt, das als kleinstes gemeinsames Vielfaches der ersten $d-1=5$ minimalen Polynome in GF(2^6) definiert ist. α sei primitives Element in GF(2^6), und die Konstruktion der Elemente α^i und der zugehörigen minimalen Polynome gelingt mit Hilfe des primitiven Polynoms $P(x) = x^6 + x + 1$. Für das erste Element $\alpha^0 = 1$ ergibt sich $m_{1(x)} = x - 1$ als minimales Polynom. Die minimalen Polynome zu α, α^2 und α^4 sind identisch:

$$m_{\alpha(x)} = x^6 + x + 1 \tag{5.1.9}$$

Das minimale Polynom zu α^3 lautet:

$$m_{\alpha^3(x)} = x^6 + x^4 + x^2 + x + 1 \tag{5.1.10}$$

Daraus folgt für das Generatorpolynom $g(x)$:

$$g(x) = m_{1(x)} \cdot m_{\alpha(x)} \cdot m_{\alpha^3(x)} \tag{5.1.11}$$

$$= x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^3 + x + 1 \tag{5.1.12}$$

Mit Hilfe dieses Generatorpolynoms ist es nun leicht möglich, eine Codierung- und Decodierungsvorschrift anzugeben.

5.1.3.3 Codierung

Es sei an dieser Stelle nochmals betont, daß eine systematische Codierung betrachtet wird. Deshalb wird zunächst das Informationswort unverändert ins Codewort übernommen. Die Erzeugung der Parity-Bits wird durch die Modulo-Division des Informationswortes $i(x)$ mit dem Generatorpolynom $g(x)$ erreicht.

$$p(x) = i(x) \cdot x^{n-k} \bmod g(x) \tag{5.1.13}$$

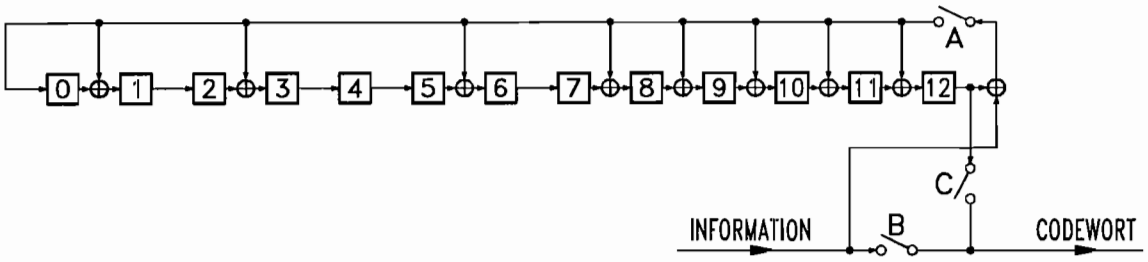


Bild 5.1.3 Encoder für (57,44,6)–BCH–Code.

Eine derartige Division läßt sich leicht durch ein rückgekoppeltes Schieberegister mit Exklusiv–Oder–Gliedern realisieren. Die EXOR–Verknüpfung entspricht einer Addition modulo 2. Das Bild 5.1.3 zeigt den schematischen Aufbau eines solchen Schieberegisters.

Die Schaltung funktioniert folgendermaßen: Am Anfang sind die Schalter A und B geschlossen und C ist offen. Das Informationswort wird nun Bit für Bit (MSB zuerst) ins Codewort geschoben. Gleichzeitig wird mit dem rückgekoppelten Schieberegister das Prüfwort gebildet. Nachdem das Informationswort durchgeschoben ist, werden die Schalter A und B geöffnet und Schalter C geschlossen, so daß nun das Prüfwort zum Informationswort ins Codewort geschoben werden kann. Das Codewort ist nun vollständig erstellt.

5.1.3.4 Decodierung

Die Decodierung ist auf Grund der Fehlerkorrektur aufwendiger als die Codierung. Die Decodierung soll hier exemplarisch an einem Meggitt–Decoder gezeigt werden. Es existieren eine Vielzahl von Varianten des Meggitt–Decoders, siehe z. B. [Bla84, S. 140-158]. Wiederum kommen rückgekoppelte Schieberegister zur Division modulo $g(x)$ zum Einsatz.

Der Meggitt–Decoder ist bei wenigen zu korrigierenden Fehlern sehr aufwandsoptimal, da er die zyklische Eigenschaft des Codes nutzt, so daß nur die Fehlermuster zu der obersten Bit–Stelle (MSB) gespeichert werden müssen.

Da ein gekürzter Code vorliegt und somit die zyklischen Eigenschaften des Codes zunächst verloren sind, muß eine vorwärts gerichtete Einkopplung $a(x)$ der empfangenen Daten–Bits vorgenommen werden, damit der Meggitt–Decoder wieder verwendet werden kann. Diese Einkopplung berechnet sich bei Kürzung um b Bit aus der Bedingung

$$s(x) \;=\; a(x)i(x) \bmod g(x) = x^{n-k+b} i(x) \bmod g(x) \tag{5.1.14}$$

Damit berechnet sich die Vorwärtseinkopplung für den gekürzten (57,44,6) Code zu

$$\begin{aligned} a(x) &= x^{63-50+6} \bmod g(x) \;=\; x^{19} \bmod g(x) && (5.1.15)\\ &= x^{19} \bmod x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^3 + x + 1 && (5.1.16)\\ &= x^{10} + x^7 + x^6 + x^5 + x^3 + x + 1 && (5.1.17) \end{aligned}$$

Der hier verwendete Code kann maximal 2 Fehler korrigieren. Da nur die Fehler in der höchsten Bit–Position detektiert werden, treten maximal $1 + \binom{56}{1} = 1 + 56 = 57$ verschiedene Fehlermuster auf, siehe Tabelle 5.1.2. Die 57 zugehörigen Syndrom–Werte sind einfach durch die Decodierung der zugehörigen Fehlermuster zu bestimmen und in [Her91a, B.3] aufgelistet.

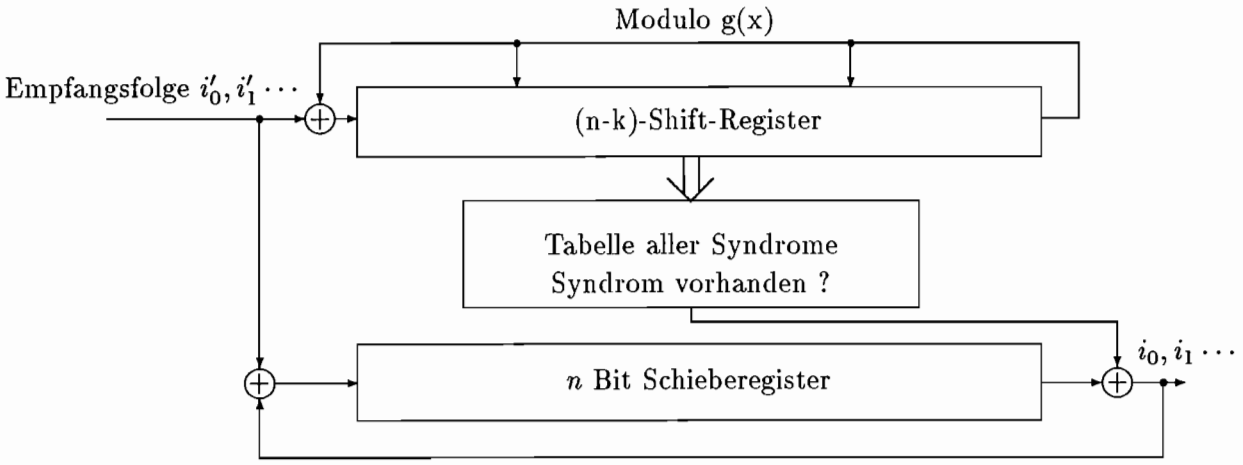


Bild 5.1.4 Schematische Darstellung eines Meggitt–Decoders.

Damit sind die Voraussetzungen zur Konstruktion des Meggitt–Decoder erklärt. Bild 5.1.5 zeigt den Aufbau des Meggitt–Decoders für den vorgeschlagenen (57,44,6)–BCH-Code.

Der Meggitt–Decoder läuft in zwei Phasen ab. In der Initialisierungsphase wird das Codewort gleichzeitig im Syndromregister modulo $g(x)$ dividiert und in das untere Schieberegister geschoben. Dazu werden 57 Taktzyklen benötigt. Nun beginnt die eigentliche Fehlerkorrektur.

Der Inhalt des Syndromregisters wird in jedem Schritt mit der Syndromtabelle verglichen. Ist das aktuelle Syndrom in der Tabelle vorhanden, so liefert der Ausgang eine 1, ansonsten eine 0. Dieses „Treffer–Bit“ wird mit der unteren Schieberegisterfolge EXOR verknüpft, so daß bei detektiertem Fehler in der obersten Stelle eine Korrektur durchgeführt wird. Gleichzeitig wird das Treffer–Bit dem Syndromregister zugeführt, damit der gerade korrigierte Fehler aus dem Syndromregister entfernt wird.

Nun wird das Syndromregister 1 Bit geschoben und modulo $g(x)$ berechnet. Ebenso wird das untere Schieberegister um eine Stelle geschoben. Mit dem so neu berechneten

Nr.	Fehlermuster
1	0 0 0 ⋯ 0 0 1
2	0 0 0 ⋯ 0 1 1
3	0 0 0 ⋯ 1 0 1
⋮	.....
56	0 1 0 ⋯ 0 0 1
57	1 0 0 ⋯ 0 0 1

Tabelle 5.1.2 Tabelle der möglichen Fehlermuster

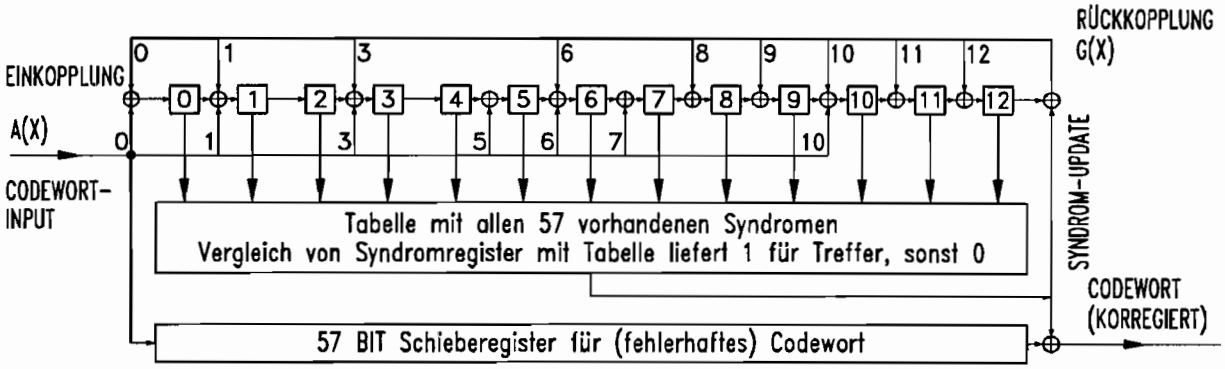


Bild 5.1.5 Meggitt–Decoder für (57,44,6)–BCH–Code.

Syndrom beginnt die oben beschriebene Auswertung der Syndromtabelle und Korrektur von vorne. Dieser Korrekturvorgang wird 57 mal wiederholt. Bei erfolgreicher Fehlerkorrektur enthält das Syndromregister zum Schluß nur Nullen. Ist das Syndromregister nicht Null, so sind mehr als 2 Fehler im Codewort vorhanden gewesen, die aber nicht korrigiert werden können.

Bei der Realisierung des Meggitt–Decoders bereitete nur die FPGA–Implementierung der Syndrom–Tabellen Schwierigkeiten, da die CLBs nur LUT mit 5 Eingangsvariablen besitzen. In einer ersten Implementierung wurden die 57 Produkterme in einer Tabelle realisiert. Hierzu wurde ein externes EPROM der Kapazität $2^{13} \times 1$ Bit benötigt. Der Syndromwert wird als 13 Bit breite Adresse an den Speicherbaustein angelegt und liefert bei den 57 Produkttermen („Treffern“) eine 1, ansonsten eine Null zurück. Für die Implementierung der Syndromtabellen mittels FPGA ergaben sich mit dem vom Xilinx–Schaltungsoptimierer (Xnfopt) durchgeführten Tests von 300 verschiedenen Komprimierungsstrategien bei 4 zu 2 bzw. 5 zu 1 LUT–Konfiguration der CLBs 163 bzw. 132 benötigte Elementarblöcke. Die in [dR93, Uni FFM, FB Informatik] durchgeführten Optimierungen mittels den „Binary Decision Diagrams“, welche die Binärfunktionen mittels eines Entscheidungsbaum darstellen, diesen Baum besonders gut komprimieren können und zuletzt optimal auf die konkrete Hardware abbilden, erbrachten wesentliche Verbesserungen bezüglich des CLB–Aufwands. Hier ergaben sich 50 bzw. 58 CLBs für die LUT–Konfigurationen 4 zu 2 bzw. 5 zu 1. Dies steht im Einklang mit den in [SVGR93] veröffentlichten Ergebnissen, bei denen ebenfalls eine BDD–Optimierung von zufällig verteilter Logik (Random–Logik) wesentliche Verbesserungen erbrachte. Bleibt zu hoffen, daß diese mächtigen BDD–Optimierungen in naher Zukunft auch in allen kommerziellen Produkten zur Optimierung von FPGA–Random–Logik eingesetzt werden.

Tabelle 5.1.3 zeigt den gesamten Realisierungsaufwand der untersuchten Decoderschaltungen.

Der (57,44,6)–BCH–Code kann auf Grund der Distanz $6 = (2 \cdot t + 1) + 1$ neben den zwei Korrekturen auch eine weitere Auslöschung beheben. Ist von Seiten des Empfängers eine Angabe der Zuverlässigkeit (Fehlerwahrscheinlichkeit) einzelner Bits möglich, so kann für das Bit mit der größten Fehlerwahrscheinlichkeit eine Auslöschung markiert werden. Somit ist eine Korrektur von 3 Bit möglich. Eventuell kann es bei vorhandener Zuverlässigkeitsangabe der Bits von Vorteil sein, wenn statt der Korrektur der Fehler *nur* Auslöschungen korrigiert werden. Mit dem vorgeschlagenen Code können dann 5

	Syndromtabellen–Realisierung		
Funktionsgruppe	EPROM	Xilinx (Xnfopt)	[dR93, Uni FFM]
PC–Interface	36	36	36
EPROM–Ansteuerung	(13+1 I/OB)	–	–
Syndromtabelle	–	132	50
64 Bit FIFO	32	32	32
Meggitt–Decoder	12	12	12
Ablaufsteuerung	21	21	21
Summe	101	233	151

Tabelle 5.1.3 Realisierungsaufwand in CLBs für die in drei Varianten des Meggitt–Decoders.

Auslöschungen korrigiert werden. Diese „Soft–Decision" kann durch systematisches Austesten aller Möglichkeiten mittels des Meggitt–Decoders erreicht werden. Üblicherweise kommt hier aber eine Korrektur im Spektralbereich zur Anwendung, deren Behandlung aber den Rahmen dieser Betrachtungen sprengen würde. Deshalb soll an dieser Stelle nur auf die umfangreiche Literatur zu diesem Thema verwiesen werden ([Bla84, S. 166-200], [Swe91, S. 81-107], [Wig88, S. 65-73]).

5.1.4 Vergleich der verschiedenen Verfahren

Im folgenden soll nun das Protokollfeldverfahren und die vorgeschlagenen Fehlercodes verglichen werden. Bei bekannter Einzelbitfehlerwahrscheinlichkeit P_B ergibt sich, falls in k Informationsstellen genau t–Fehler auftreten, für die Gesamtfehlerwahrscheinlichkeit:

$$P(t) = \binom{k}{t} P_B^t (1 - P_B)^{k-t} \tag{5.1.18}$$

Die Wahrscheinlichkeit für ein fehlerfreies Minutenprotokoll ohne Fehlerkorrektur ergibt sich zu

$$P_M(0) = (1 - P_B)^{44} \tag{5.1.19}$$

Vernachlässigt man die Mehrdeutigkeit bei Stunden/Datumswechsel, so wird durch die Protokollfeldauswertung, wie in Abschnitt 5.1.1 gezeigt, bei L minütiger Auswertung, wobei L möglichst ungerade sein sollte, eine Korrektur von $\lfloor L/2 \rfloor$ Fehlern erreicht. Die Einzelbitfehlerwahrscheinlichkeit verbessert sich somit auf P_{PB}:

$$P_{PB}(L) = \sum_{j > \lfloor L/2 \rfloor}^{L} \binom{L}{j} P_B^j (1 - P_B)^{L-j} \tag{5.1.20}$$

Die Wahrscheinlichkeit für ein richtiges Minutenprotokoll bei Auswertung von L Minuten ergibt sich zu

$$P_{MP} = (1 - P_{PB}(L))^k = (1 - P_{PB}(L))^{44} \tag{5.1.21}$$

Für einen t–Fehler korrigierenden Code ergibt sich dann ein korrektes Minutentelegramm, wenn weniger als t Fehler im Protokoll auftreten. Für die Wahrscheinlichkeit eines fehlerfreien Minutentelegramms P_{MC} folgt

$$P_{\mathrm{MC}} = \sum_{j=0}^{t} P(j) \tag{5.1.22}$$

wobei

$$P(j) = \binom{n}{j} P_{\mathrm{B}}^{j} (1 - P_{\mathrm{B}})^{n-j} \tag{5.1.23}$$

gilt. Es ist zu beachten, daß im Gegensatz zu Gleichung (5.1.18) nun als Anzahl die gesamte Codewortlänge n d. h. 44+13=57 benutzt werden muß.

Bild 5.1.6 zeigt den Vergleich der Fehlerwahrscheinlichkeiten von Protokollfeldverfahren und t–Fehlerkorrektur. Da für das Protokollfeldverfahren nur 44 Bit — hingegen für die t-Fehlerkorrektur 57 Bit — berücksichtigt werden müssen, erfolgt eine Umrechnung über die Wahrscheinlichkeit für ein fehlerfreies Minutenprotokoll (P_{MP} bzw. P_{MC}). Gezeigt sind jeweils die zu erwartenden Einzelbitfehlerwahrscheinlichkeiten vor bzw. nach der durchgeführten Korrekturmaßnahme. Bild 5.1.6 macht deutlich, daß für große Werte der

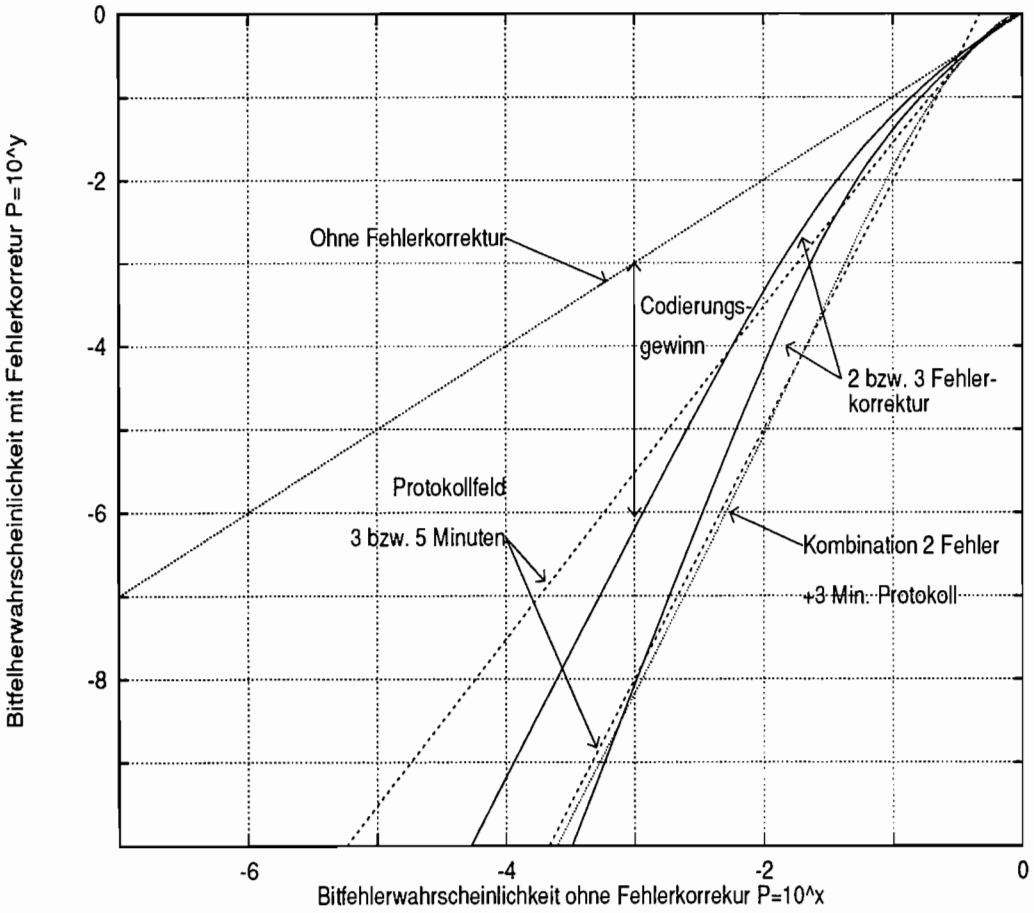


Bild 5.1.6 Vergleich der Bitfehlerwahrscheinlichkeiten.

Einzelbitfehlerwahrscheinlichkeit die t–Fehlerkorrektur eine geringfügige Verschlechterung bedeutet, hingegen bei kleineren Werten von P_B eine $(t = 2)$–Fehlerkorrektur erheblich bessere Ergebnisse liefert als eine 3 minütige Protokollfeldauswertung. So ergibt sich z. B. bei $P_B = 10^{-3}$ ohne Korrektur mit der vorgeschlagenen $(t = 2)$–Fehlerkorrektur einen Vergleichswert von $P_B = 10^{-6}$, d. h. eine Verbesserung um den Faktor 1000 gegenüber der Auswertung ohne Fehlerkorrektur.

Zum Vergleich von Protokollfeldauswertung und t–Fehlerkorrektur ergibt sich aus Bild 5.1.6, daß für 3 bzw. 5 Minuten Protokollfeldauswertung und 2 bzw. 3 Fehlerkorrektur vergleichbare Codierungsgewinne gegenüber der Auswertung ohne Fehlerkorrektur erreicht werden. Da, wie in Abschnitt 5.1.3 gezeigt, bei der t–Fehlerkorrektur für die zu realisierende Hardware (oder Software in einem MC/DSP) nur wenige und einfache logische Verknüpfungen berechnet werden müssen, ist der anzusetzende Komplexitätszuwachs, Zeit– und Leistungsverbrauch minimal. Wird die Zeitverzögerung beim Einschalten des Geräts zum Aufbau des Protokollfeldes vernachlässigt, so ist der Komplexitäts– und Zeitbedarf für die Korrekturmaßnahme durch Protokollfeldauswertung ähnlich gering anzusetzen wie für die t–Fehlerkorrektur. Hingegen benötigt man für 3 bzw. 5 Minuten Protokollfeld die 3 bzw. 5 fache Leistung, da im allgemeinen der Leistungsverbrauch des Analogteils weit höher liegt als der des Digitalteils. Zusammenfassend kann man festhalten, daß die t–Fehlerkorrektur bei vergleichbaren Codierungsgewinnen bezüglich des Leistungsbedarfs wesentliche Vorteile gegenüber der Protokollfeldauswertung besitzt.

Ausblickend sei noch kurz die Kombination von Protokollfeldverfahren und t–Fehlerkorrektur angesprochen. Trägt man die empfangenen Minutenprotokolle in eine Matrix ein, so erhält man für die 44 Bit Zeitinformation einen **Produktcode**. Aufgrund des Minuteninkrements kann leider in den Prüfstellen 0-14 über die Minuten *kein* Wiederholungscode angesetzt werden, so daß die gesamte Produktcodematrix über alle Bits nicht **vollständig** ist. Besitzen die Codes minimale Distanzen d_1 bzw. d_2 und treten nur Fehler im Bereich der 44 Zeit–Bits auf, so können mit einem geeigneten Algorithmus[§] bis zu $\lfloor (d_1 \cdot d_2 - 1)/2 \rfloor$ Fehler korrigiert werden, siehe z. B. [Swe91, S. 144], [Bla84, S. 293]. Bei 3 Minuten Protokollfeld $(d_1 = 3)$ und dem vorgeschlagenen (57,44,6)–BCH–Code $(d_2 = 6)$ können so bis zu $\lfloor (3 \cdot 6 - 1)/2 \rfloor = 8$ Fehler korrigiert werden.

Ein stark vereinfachter Algorithmus würde zunächst eine Protokollfeldauswertung durchführen und danach die Fehlerkorrektur mit dem BCH–Code benutzten. Für diesen einfachen Algorithmus bei 3 Minuten Protokollfeld und $(t = 2)$–Fehlerkorrektur ist die resultierende Bitfehlerwahrscheinlichkeit ebenfalls in Bild 5.1.6 eingetragen.

Eine weitere Verbesserung könnte, wie bereits im Abschnitt 5.1.3.4 für eindimensionale Codes angesprochen, durch die Auswertung einer eventuell vorhandenen Zuverlässigkeitsangabe der einzelnen Bits der Produkt–Code–Matrix erreicht werden. Dabei würde allerdings der Decodierungsaufwand durch „Soft–Decision–Decoding“ gegenüber der Decodierung ohne Soft–Decision weiter ansteigen. Eine Aussage über den Codierungsgewinn ist bei „Soft–Decision–Decoding“ dann nur mit Hilfe von Rechnersimulationen möglich.

[§] Hier ist ein mehrfach durchzuführende Fehler– und Auslöschungskorrektur nötig, die eine vereinfachte Form des „Generalized Minimum Distance Decoding“ (GMD) [Bla84, S. 466] darstellt.

5.2 Personenrufsystem für Langwelle

Als Zusatzgerät zu mobilen oder festinstallierten Telefonen sind „Personenrufsysteme“ (Paging–Systeme) ideal. Ein solch kleiner, stift– oder armbanduhrgroßer Empfänger zeigt an, daß ein Teilnehmer telefonisch erreicht werden soll [Ker89].

Eins der ersten in Deutschland, Frankreich und der Schweiz verfügbaren Personenrufsystemen ist das „Eurosignal“. Hier kann einer Person eine Nachricht in Form einer Nummer übermittelt werden. Der Personenruf wird durch das Anwählen einer bestimmten Telefonnummer ausgelöst. Eine komfortablere Übertragung ist mit dem „Cityruf–System“ der Telekom möglich. Hier können verschiedene Töne, rein numerische oder auch bis zu 80 alphanumerische Zeichen ausgesendet werden. Die gewählte Trägerfrequenz (466 MHz) bedingt, daß die Aussendung einer Sendeanlage auf Ballungszentren von Großstädten beschränkt ist.

Weitere geplante bzw. im Aufbau befindliche Systeme verwenden Satellitentechnik (Inmarsat), oder mobile Datenkommunikationsnetze (Modacom), die allerdings weitaus aufwendigere Empfangstechnik benötigen. Das einzige für LW verfügbare Personenrufsystem der Schweiz (HBG–Sender) gestattet nur einen Personenruf alle zwei Minuten [Hil93, S. 49].

In dem im folgenden beschriebenen Personenrufsystem wurden konsequent die Vorteile der LW–Technik bei unidirektionaler Übertragung eingesetzt. Der „Cityruf“ ist auf kleine Bereiche (ca. 50 km von der Sendeanlage) beschränkt, wohingegen mittels Langwelle mit einem Sender weite Teile Mitteleuropas abgedeckt werden können. Zudem besitzen LW beim Eindringen in Gebäuden (und Kleidung) wesentlich bessere Ausbreitungseigenschaften als UHF–Wellen [Hil93, S. 26-39]. Außerdem müssen bei Cityruf Sender innerhalb einer Zone genau synchron laufen, damit es zu keinen Auslöschungen durch Interferenz kommt. Zahlreiche Sendestationen sind für eine flächendeckende Versorgung nötig, so z. B. ca. 1000 Sendeanlagen bei Modacom–System für Deutschland. Darüber hinaus ist beim „Cityruf“ keine Nachrichtenverschlüsselung vorhanden, so daß Nachrichten leicht abgehört oder sogar falsche Nachrichten ausgesendet werden können. Weiterhin erscheint die Wahl eines Blockcodes zur Fehlerkorrektur, wie er beim Cityruf verwendet wurde, für eine kontinuierliche Datenverarbeitung hinderlich [Dru89], [Tel91].
Das entworfene Pagingsystem ist durch folgendes Pflichtenheft gekennzeichnet:

- Teilnehmerzahl ca. 4000000
- Maximale Wartezeit je Funkruf 15 min
- im Mittel 1 Anruf pro Teilnehmer je Tag
- geringe Leistungsaufnahme des Empfängers
- Maßnahmen zur Fehlerkorrektur
- Übertragung alphanumerischer Nachrichten
- Datenverschlüsselung

Neben der vollständigen Erfüllung des Pflichtenheftes sind folgende erzielte Leistungsmerkmale zu nennen:

- Reichweite des Senders größer als 1000 km (LW–Sender, 50 kW, 3000 Baud)
- 3 Nachrichtentypen
- Übertragung von Datum und Uhrzeit
- Adressierung von 144 Gruppen à 2^{16} Teilnehmern
- Systematischer Faltungscode ($J = 8$ Trial–and–Error–Code mit Majority–Decoder)
- DES basierte Strom–Chiffre zur Verschlüsselung

Im folgenden wird zunächst das entworfene Datenformat vorgestellt. Danach erfolgt die Beschreibung der Fehlerkorrekturschaltung und des verwendeten Verschlüsselungsalgorithmus. Abschließend wird noch das realisierte Gesamtsystem kurz erläutert.

5.2.1 Beschreibung des Datenformats

5.2.2 Das Minutenintervall

Durch einen festen Zeitbezug des verwendeten Datenformats ist es zum einen möglich, die Teilnehmer in Gruppen zu unterteilen, so daß ein Empfänger nur einen kleinen Teil der gesamten Übertragungszeit „auf Empfang" sein muß. Zum anderen ermöglicht es durch wenige zusätzliche Bits die Übertragung von Datum und Uhrzeit.

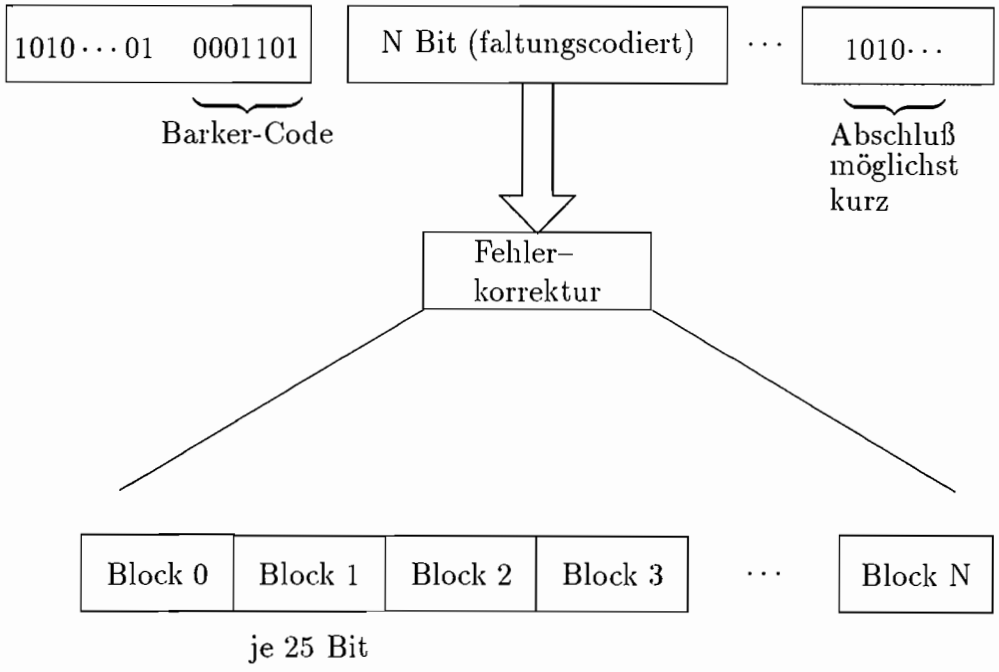


Bild 5.2.1 Aufbau des Sekundenintervalls.

Den prinzipiellen Aufbau des Datenformats zeigt Bild 5.2.1. Zunächst sind einige wenige Bit zur Synchronisation vorgesehen. Danach folgen die mit einem systematischen Faltungscode zur Fehlerkorrektur versehenen eigentlichen Daten– und Adreßblöcke *à* 25 Bit. Zum Abschluß sind einige Stopf–Bits zur Vervollständigung des Rahmens vorgesehen.

Zur Synchronisation wurde anders als bei „Cityruf“ keine 512 Bit lange Sequenz benutzt, sondern eine kurze $01\cdots$ Folge, um die Laufzeitunterschiede innerhalb Deutschland auszugleichen. Eine kurze Barker–Sequenz dient zur Kennzeichnung des Blockanfangs. Diese kurze Synchronisationssequenz wird im Sekundentakt des DCF77 Signals ausgestrahlt. Dem etwas erhöhten Aufwand für die Realisierung der zweiten Empfangsfrequenz steht der Vorteil einer nicht benötigten, fortwährenden 512–Bit Korrelation zur Synchronisation gegenüber. Wegen der geringeren Verlustleistung erscheint die gewählte Synchronisation der Synchronisation des „Cityrufs“ überlegen.

Die Gruppeneinteilung wurde hierarchisch durchgeführt. Die oberste Gruppeneinteilung findet auf Minutenebene statt. Es gibt 12 Gruppen, so daß jede dieser 12 Gruppen genau 5 mal pro Stunde angesprochen wird, siehe Pflichtenheft. Diese Minutengruppen sind wiederum in 12 Gruppen eingeteilt, so daß ingesamt 144 Gruppen durch den Zeitbezug unterschieden werden können, ohne daß zusätzliche Adreß–Bits übertragen werden müssen. Weiterhin bietet eine Gruppeneinteilung die Möglichkeit, die Adressen einzelner Gruppen nicht vollständig zu vergeben, so daß (bei erhöhten Kosten für den Funkruf) Teilnehmer häufiger erreichbar sind als andere Teilnehmer niedrigerer Priorität.

Die Nachrichten selbst sind folgendermaßen aufgebaut: Zunächst einmal sind 2 Bit für 3 Typen der „Art der Nachricht“ vorgesehen. Dies sind Adresse mit Status–Bits, numerische oder alphanumerische Nachrichten. Ein Adreßwort besitzt dabei in der ersten Bit–Stelle eine Eins, dann 16 Bits für die Adresse, zwei Bits für „Art der Nachricht“ und 6 Status–Bits[†], siehe Bild 5.2.2. Die Datenworte ihrerseits beginnen mit einer Null, gefolgt von 6 numerischen oder 4 alphanumerischen Zeichen, siehe Bild 5.2.3 und 5.2.4.

Außerdem ist die Verschlüsselung der Datenworte vorgesehen. Es liegt in der Natur des Systems, daß anders als bei vielen Kryptosystemen der Schlüssel nachträglich nicht verändert werden kann, ohne daß die Veränderung im voraus bestimmt ist oder über den Kanal übertragen wird. Die Übertragung eines Schlüssels über den Kanal kann nicht in verschlüsselter Form geschehen, da nicht sichergestellt werden kann, daß jeder Empfänger

[†] Die Status–Bits sind als minimale Zusatzinformation vorgesehen, wenn der Übertragungskanal stark ausgelastet ist und keine zusätzlichen Datenworte übertragen werden können. Von den 6 Bit dienen 4 Bit der Kennzeichnung des Rückrufs über die Wahlspeichertasten auf dem eigenen Telefon und 2 Bits der Dringlichkeit der Nachricht bzw. des Rückrufs.

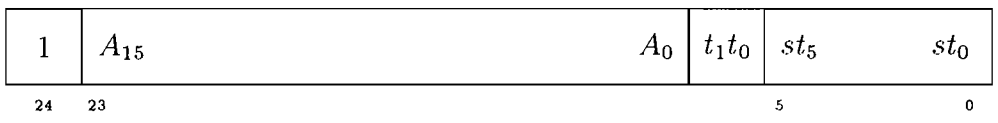


Bild 5.2.2 Aufbau des Adreßworts.

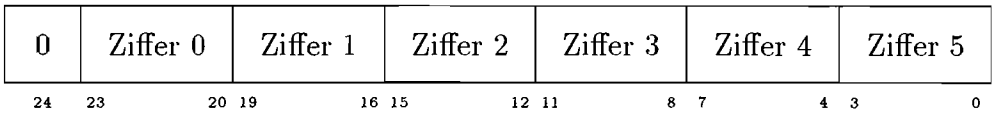


Bild 5.2.3 Das numerische Datenwort.

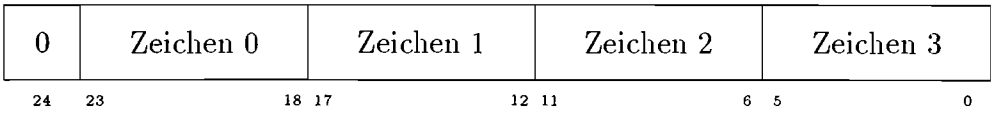


Bild 5.2.4 Das alphanumerisches Datenwort.

zum Zeitpunkt der Schlüsseländerung betriebsbereit ist. Da aber alle invarianten Teile des Verschlüsselungsverfahrens potentiell einem Angreifer zugänglich sind (durch Kryptoanalyse), wird hier für jeden Empfänger ein individueller Schlüssel vorgesehen, der vom Sender in einer Tabelle gespeichert wird und in jedem Empfänger fest codiert ist. Dieser Schlüssel wird durch einen weiteren Teilschlüssel ergänzt, der aus der übertragenen Zeitinformation besteht.

Gesamtschlüssel:

Zeit–Teil	Adreßteil

5.2.3 Fehlerkorrekturschaltung

Greift man die ausführliche Diskussion der Unterschiede zwischen Faltungs– und Blockcodes aus Abschnitt 5.1.3 auf, so erscheint es unmittelbar einsichtig, daß für eine kontinuierliche Verarbeitung des Bit–Stroms und den damit verbundenen geringen Hardware–Aufwand und Leistungsbedarf eine Faltungscode geeigneter erscheint. Im folgenden sollen daher leistungsfähige und aufwandsgünstige *Decoderschaltungen* untersucht werden, da die senderseitige *Encoderschaltung* lediglich einen einmaligen Aufwand bedeutet. Es wird dabei kein Wert auf eine vollständige Darstellung und Erläuterung aller möglichen Decoderschaltungen gelegt — sondern es wird lediglich eine Suche nach günstigen Verfahren durchgeführt, so daß Decoderschaltungen, die unmittelbar ungünstig für eine Paging–Empfängerrealisierung erscheinen, nicht untersucht werden.

Bild 5.2.5 zeigt eine übliche Darstellung für Faltungscoder, an der nochmals die Begriffe Informationsblocklänge bzw. Symbollänge k, Codewortlänge n und Einflußlänge $\nu = m \cdot k$ deutlich hervortreten. Im speziellen zeigt Abbildung 5.2.5 einen Faltungscoder mit $k = 3$, $n = 5$, $m = 7$ und $\nu = 21$. Für diesen (n, k)–Baumcode ergibt sich eine Coderate von $R = k/n = 0,6$.

Üblicherweise werden vier grundlegende Kriterien zur Klassifizierung verschiedener Baumcodes benutzt. Dies sind die **endliche Einflußlänge**, die **Zeitinvarianz**, die **Li-**

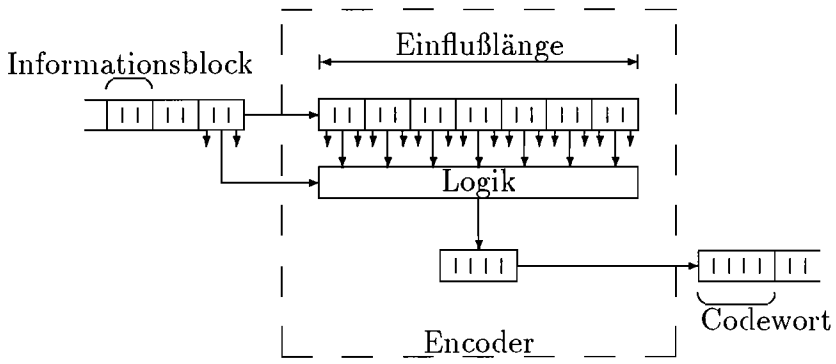


Bild 5.2.5 Veranschaulichung eines Schieberegister–Encoders.

narität und die Frage, ob der Code **systematisch** ist. Die Begriffe wurden bereits im Abschnitt 5.1.3 erläutert. Abbildung 5.2.7 zeigt eine systematische Darstellung gebräuchlicher Baumcodes anhand dieser vier Kriterien [Bla84, S. 351].

5.2.3.1 Der Maximum Likelihood Sequence Decoder — Viterbi–Decoder

Bereits bei der gedächtnisbehafteten Demodulation von FM–Signalen im Abschnitt 4.3.3 wurde die Sequenzschätzung mittels Viterbi–Decoder kurz angesprochen. Der Viterbi–Decoder liefert bei einer (fehlerbehafteten) Empfangssequenz die zugehörige Sendesequenz mit der geringsten Hamming–Distanz. Da er somit den optimalen Weg durch ein Trellis–Diagramm findet, ist der Viterbi–Algorithmus bei einer gedächtnislosen Rauschstörung ein *optimaler* Sequenzschätzer (MLSE). Als Vorteile des Viterbi–Decoders sind die konstante Decodierzeit und das MLSE–Prinzip zu nennen — als Nachteile der große Speicherbedarf und damit die Einschränkung auf Codes mit kleiner Gedächtnislänge.

Da die im Abschnitt 5.2.3.2 beschriebenen sequenziellen Decoder auf dem MLSE–Prinzip des Viterbi–Algorithmus aufbauen, erfolgen an dieser Stelle noch einige Grundüberlegungen. Lineare Baum–Codes können durch ein **Trellis–Diagramm** dargestellt werden, da sich die einzelnen Zustände im Code–Baum zyklisch wiederholen. Ein Baum–Code mit der Einflußlänge $\nu = m \cdot k$ kann durch die 2^ν verschiedenen Zustände des Schieberegisters, siehe Bild 5.2.6, charakterisiert werden. Das Trellis–Diagramm besitzt bei einer Code–Rate $R = k/n$ je Zustand bzw. **Knoten** 2^k ankommende und 2^k wegführende **Äste**. An den Ästen werden die Codesymbole der Länge n notiert. Bei binären Trellis ($k = 1$) ist es üblich, eine Verzweigung nach oben einer übertragenen Null und eine Ver-

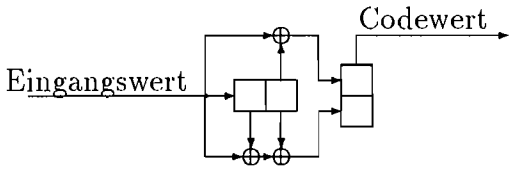


Bild 5.2.6 Encoder für einen $R = 1/2$–Faltungscode.

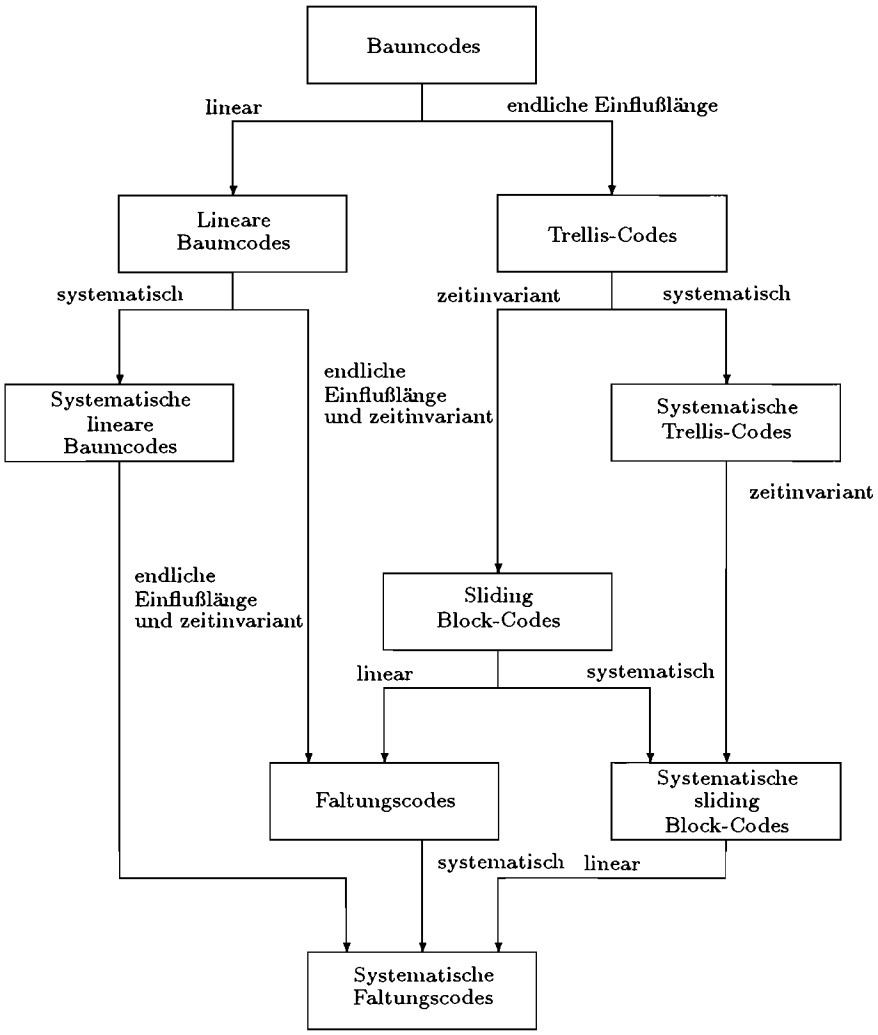


Bild 5.2.7 Hierarchische Einteilung von Baumcodes.

zweigung nach unten einer übertragenen Eins zuzuordnen. Dem Encoder aus Bild 5.2.6 ist das Trellisdiagramm aus Abbildung 5.2.8 zuzuordnen.

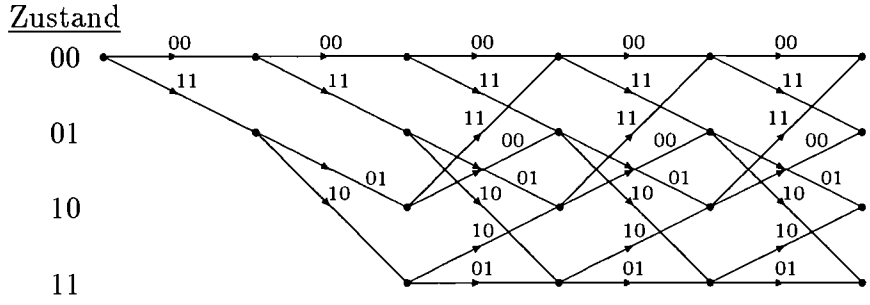


Bild 5.2.8 Trellisdiagramm für $R = 1/2$–Faltungscodes.

Für die MLSE–Decodierung reicht es völlig aus, immer nur die 2^{ν}–Pfade und deren Metriken, die an den Knoten einer Stufe ankommen, zu speichern, da letztendlich der MLSE–Pfad einen dieser Knoten beinhaltet. Alle Pfade, die bei einem Knoten ankommen und eine kleinere Metrik als die des „Survivor" besitzen, brauchen nicht mehr gespeichert zu werden, da diese Pfadstücke niemals zum MLSE–Pfad gehören können und man gewöhnlich nur den Pfad durch das Netz bestimmen will, der zuletzt die größte Metrik besitzt. Dennoch kann es natürlich vorkommen, daß der Knoten, der zu einem gewissen Zeitpunkt als Endpunkt des maximalen Pfades gilt, später nicht mehr zum MLSE–Pfad gehört. Dieses „Überstimmen" einer lokalen Falschdetektion ist eine Analogie zu der im Abschnitt 4.3.3 beschriebenen gedächtnisbehafteten Demodulation mit großen lokalen Störimpulsen. Decodierungbeispiele zum Viterbi–Algorithmus sind z. B. in [Dor93, S. 152], [Bla84, S. 381] oder [Swe91, S. 120-3] zu finden.

Aufgrund der Forderung eines endlichen Pfadspeichers und der Verzögerungszeit durch das Decodieren können die 2^{ν} Pfade nicht beliebig lang untersucht werden. Simulationsergebnisse zeigen, daß die Berücksichtigung der 4-5 fachen Gedächtnislänge im Pfadspeicher nur zu einer unerheblichen Degradation gegenüber einem unendlich langen Pfadspeicher führt [Dor93, S. 151]. Decodiert wird üblicherweise durch Rückverfolgung zum Pfadanfang des Pfades, der zu einem bestimmten Zeitpunkt die maximale Metrik besitzt. Der nötige Schaltungsaufwand ist in den Bildern 5.2.9 und 5.2.10 veranschaulicht.

Die Hardware des Viterbi–Decoder läßt sich entsprechend den Abbildungen 5.2.9 und 5.2.10 in drei Teile untergliedern: **Pfadspeicher mit Decoderausgang, Survivor–Berechnung** und **Maximumbestimmung** . Als *Pfadspeicher* werden $4 \cdot \nu \cdot 2^{\nu}$ Bit und halbsoviele CLBs benötigt. Für den *Decoder zum Ausgang* werden

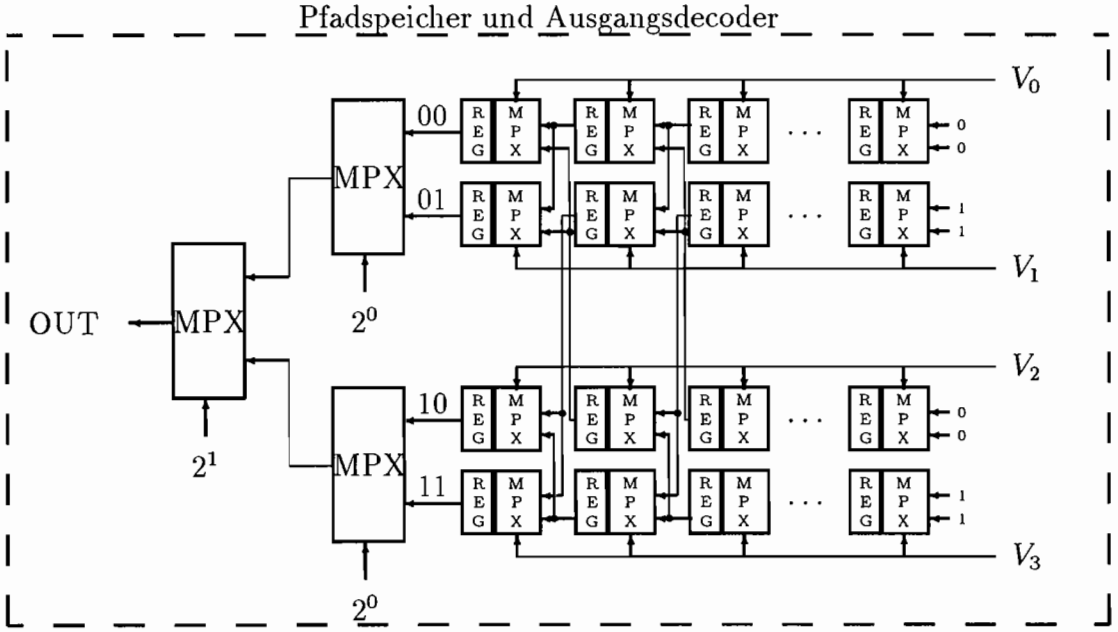


Bild 5.2.9 Viterbi–Decoder mit Pfadlänge $4 \cdot \nu$ und $2^{\nu=2}$ Knoten. (a) Pfadspeicher und Ausgangsdecoder.

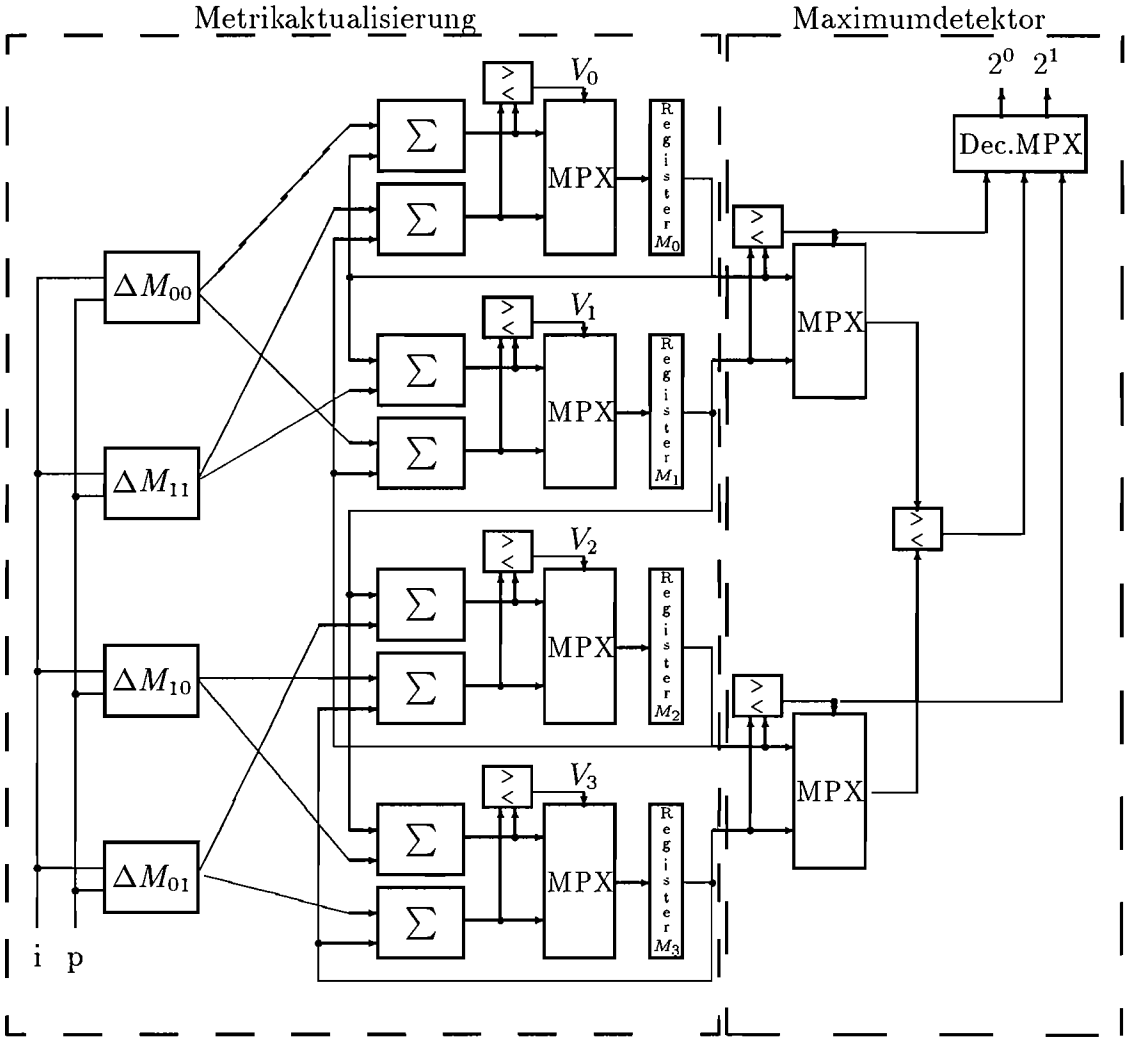


Bild 5.2.10 Viterbi–Decoder mit Pfadlänge $4 \cdot \nu$ und $2^{\nu=2}$ Knoten. (b) Metrikberechnung des Viterbi–Decoder.

$(1 + 2 + \ldots 2^{\nu-1})$ an 2 zu 1 MPX benötigt. Für die Survivor–Berechnung sind Metrikadditiv und Metrikclear, Metriksumme, Vergleicher je mit $(\lceil \log_2(\nu * n) \rceil + 1)$ Bit Breite und der Metrikspeicher mit vorgeschaltetem 2 zu 1 Multiplexer vorzusehen. Für die *Maximumbestimmung* werden Vergleicher, 2 zu 1 Multiplexer und ein Decoder (der die MPX–Ansteuerung der Maximumbestimmung in die Ansteuerung für den Ausgangsdecoder umsetzt) benötigt.

Andere Werte als $k = 1$ erscheinen für eine FPGA–Implementierung noch zu aufwendig. Für $n > 2$ wäre die Informationsmenge pro Zeiteinheit zu gering, so daß im folgenden nur Codes der Rate $R = 1/2$ betrachtet werden. Die Tabelle 5.2.1 zeigt den Realisierungsaufwand für die Einflußlängen $\nu = 2$ bis 4 und beliebige ν bei $R = 1/2$.

Die Ergebnisse aus Tabelle 5.2.1 zeigen, daß aufgrund des exponentiellen Anstiegs des Aufwandes mit der Einflußlänge, ν möglichst klein gehalten werden muß. Da aber eine MLSE–Decodierung erfolgt, ist zu erwarten, daß trotz der geringen Anzahl an Fehlern,

Funktionsgruppe	$\nu = 2$	$\nu = 3$	$\nu = 4$	ν
Pfadspeicher	16	48	128	$4 \cdot \nu \cdot 2^{\nu-1}$
Decoder zum Ausgang	1,5	3,5	6,5	$1 + 2 + \ldots + 2^{\nu-2}$
Metrikadditiv	4	4	4	4
Metrikclear	1	2	4	$\lceil (2 + 4 + \ldots + 2^{\nu-1})/4 \rceil$
Metriksumme	24	64	128	$(\lceil \log_2(n\nu) \rceil + 1) \cdot 2^{\nu+1}$
Survivor–MPX	6	24	48	$(\lceil \log_2(n\nu) \rceil + 1) \cdot 2^{\nu-1}$
Metrikvergleicher	6	24	48	$(\lceil \log_2(n\nu) \rceil + 1) \cdot 2^{\nu-1}$
Maximumvergleich	4,5	14	30	$(\lceil \log_2(n\nu) \rceil + 1) \cdot \frac{1}{2} \cdot (1 + 2 + \ldots 2^{\nu-1})$
MPX	3	12	28	$(2 + \ldots + 2^{\nu-1}) \cdot \frac{1}{2} \cdot (\lceil \log_2(n\nu) \rceil + 1)$
Decoder	1	2	4	$\lceil (2 + 4 + \ldots + 2^{\nu-1})/4 \rceil$
Steuerwerk	4	4	4	
Summe:	67	197,5	428,5	

Tabelle 5.2.1 Realisierungsaufwand in CLBs von $R = 1/2$ Viterbi–Decodern für beliebige ν und berechnete Werte für $\nu = 2$ bis 4.

die in einem kleinen Fenster korrigierbar ist, sich insgesamt doch akzeptable Korrekturleistungen ergeben.

Die Auswahl geeigneter Generatorpolynome gestaltet sich folgendermaßen: Vergleicht man die in der Literatur genannten Korrektureigenschaften von systematischen und nichtsystematischen Faltungscodes, so zeigt sich, daß bei gleicher Einflußlänge *nichtsystematische* Faltungscodes bessere Korrektureigenschaften besitzen als *systematische* Faltungscodes, siehe z. B. [Pro83, S. 306-8], [Joh75, S. 465], [CC88, S. 402-7] [Bla84, S. 367] oder [Dor93, S. 162-4]. Die Verwendung eines nichtsystematischen Codes widerspricht aber der Forderung nach geringem Stromverbrauch. Dies entstammt folgender Idee: Der Empfänger soll ohne Fehlerkorrektur das eigene Adreßwort (mit bis zu t Fehlern) im Datenstrom erkennen können und erst dann die Fehlerkorrektur hinzuschalten, um etwaige Fehler (Anzahl kleiner t) zu korrigieren.

„Quick Look In" (QLI) Codes [MC71] sind eine Klasse von nichtsystematischen Faltungscodern der Rate $R = 1/2$ dergestalt, daß hier die für den Viterbi–Decoder bedeutsame *freie Distanz* ebensogute Werte liefert, wie die besten bekannten Faltungscodes mit gleicher Einflußlänge $\nu = 2$ bis 4. Vorteilhaft ist, daß die Informationsfolge durch einfache EXOR–Verknüpfung rekonstruiert werden kann. Die freie Distanz der verwendeten QLI–Codes ergibt sich für $\nu = 2, 3$ und 4 zu $d_f = 5, 6$ bzw. 7 [Joh75, S. 465].

Dieser gute Kompromiß zwischen geringem Aufwand für die Rückgewinnung der Informationsfolge und den guten Korrektureigenschaften der QLI–Codes lassen einen Einsatz dieser Codes im Paging–System sinnvoll erscheinen. Daher sind in der Tabelle 5.2.2 genannten Generatorpolynome für QLI–Codes angegeben.

Für die Berechnung der **Bitfehlerwahrscheinlichkeit** des MLSE–Decoders sind aus der Literatur zwei Verfahren bekannt:

Code	O1 = 7 O2 = 5	O1 = 74 O2 = 54	O1 = 66 O2 = 46
Einflußtiefe	$\nu = 2$	$\nu = 3$	$\nu = 4$
Distanz	Gewichtskoeffizienten w_j		
0-4	0	0	0
5	1	0	0
6	4	2	0
7	12	7	4
8	32	18	12
9	80	49	26
10	192	130	74
11	448	333	205
12	1024	836	530
13	2304	2069	1369
14	5120	5060	3476
15	11264	12255	8470
16	24576	29444	19772
17	53079	64183	43062
18	109396	126260	83346
19	103665	223980	147474
20	262144	351956	244458

Tabelle 5.2.2 Gewichtskoeffizienten für das „Union Bound" des Viterbi–Decodern mit $\nu = 2$ bis 4 für den QLI–Faltungsdecoder.

a) Berechnung mittels FSM–Darstellung und Signalsflußgraphfunktion $T(D, L, N)$ [Bla84, S. 442-4]

b) Berechnung mittels Union Bound [CC88, S. 24 und S. 244]

Das Verfahren nach **a)** ist weniger empfehlenswert, da die Bestimmung von $T(D, L, N)$ per Hand aus dem Signalflußgraphen, selbst bei kurzen Einflußlängen, recht mühsam ist. Das Verfahren nach **b)** erlaubt den Einsatz eines Digitalrechners.

Ohne Einschränkung der Allgemeinheit kann man zur Bestimmung des „Union Bounds" nur die Distanz zum Nullwort betrachten, da lineare Faltungscodes vorliegen. Beim MLSE–Decoder wird dann eine falsche Entscheidung getroffen, wenn eine Sequenz, die im Nullzustand startet und in j Stellen zum Nullwort differiert, mindestens $j/2$ Einsen besitzt. Für diese Wahrscheinlichkeit gilt:

$$P_j = \begin{cases} \sum_{i=(j+1)/2}^{j} \binom{j}{i} p^i q^{j-i} & \text{bei j ungerade} \\ \frac{1}{2}\binom{j}{j/2} p^{j/2} q^{j/2} + \sum_{i=j/2+1}^{j} \binom{j}{i} p^i q^{j-i} & \text{bei j gerade} \end{cases} \tag{5.2.1}$$

Berechnet man nun für einen Code die Anzahl w_j solcher Pfade mit dem Gewicht j, so kann für einen Code der Rate $R = 1/2$ die resultierende Einzelbitfehlerwahrscheinlichkeit mit Korrektur mittels eines kurzen Programms [MS93, Anhang C.4] zur Bestimmung der Gesamtgewichte der Eingangssequenzen w_j, berechnet werden, wobei gilt

$$P_b < \frac{1}{k} \sum_{j=0}^{\infty} w_j P_j \tag{5.2.2}$$

Die Tabelle 5.2.2 zeigt die berechneten Gewichtskoeffizienten w_j. Da P_j exponentiell für große j gegen Null strebt, müssen nur wenige w_j berechnet werden. Die zugehörigen Kurven für die Bitfehlerwahrscheinlichkeit (Bild 5.2.13) werden in Abschnitt 5.2.3.4 noch näher erläutert.

5.2.3.2 Sequentielle Decoder

Der wesentliche Nachteil einer FPGA–Implementierung des Viterbi–Decoders liegt in dem exponentiell mit der Einflußlänge ν wachsenden Speicheraufwand. Ziel der **sequentiellen Decoder** ist es nun, gerade diesen Speicheraufwand stark zu reduzieren, ohne daß ein wesentlicher Decodierungsverlust entsteht. Dies wird dadurch erreicht, daß nur ein Pfad (Fano–Decoder) oder einige wenige Pfade (Stack–Decoder) in einem Kellerspeicher abgelegt werden.

Beim **Fano–Decoder** wird nur ein Pfad in begrenzter Pfadlänge gespeichert. Die Suche des MLSE–Pfads wird dadurch gesteuert, daß die Metrik fortwährend mit einem **laufenden Schwellwert** T verglichen wird. Tritt nun aufgrund eines Übertragungsfehlers der Fall auf, daß der Metrikwert unter den Schwellwert T fällt, so wird ein neuer Pfad mit größerer Metrik gesucht. Dabei wird einen oder mehrere Knoten im Trellis zurückgegangen und versucht, mit einer anders gewählten Verzweigung eine größere Metrik zu erhalten.

Der Vorteil des Fano–Decoders ist darin zu sehen, daß auch Faltungscodes größerer Länge verwendet werden können, die üblicherweise bessere Korrektureigenschaften bieten als Decoder kurzer Einflußlängen. Obwohl der Fano–Decoder kein MLSE–Decoder ist und folglich eigentlich schlechter in der Codeausnutzung als ein Viterbi–Decoder ist, so ist es in einer praktischen Realisierung dennoch möglich, bessere Korrekturwerte zu erbringen, da ja bedeutend größere Einflußlängen verarbeitet werden können.

In Tabelle 5.2.3 ist eine CLB–Aufwandsabschätzung für den Fano–Decoder für verschiedene Einflußlängen ν gezeigt.
Als Nachteile des Fano–Decoders sind zu nennen:

- Muß im Fehlerfall mehr Knoten zurückgegangen werden als der Pfadspeicher lang ist, so sind die im Pfadspeicher stehenden Daten unbrauchbar
- Der Decoder ist nicht in der Lage, auf einen beliebigen Anfangszustand zu synchronisieren

Funktionsgruppe	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	$\nu = 6$	$\nu = 7$	$\nu = 8$	$\nu = 9$	$\nu = 10$
Pfadspeicher	10	15	20	25	30	35	40	45	50
Metrik	7	8	8	8	9	9	9	9	9
Encoder	6	9	12	15	18	21	24	27	30
Addierer	14	16	16	16	18	18	18	18	18
Vergleicher	7	8	8	8	9	9	9	9	9
Steuerwerk	5	5	5	5	5	5	5	5	5
Summe	49	61	69	76	89	97	105	113	121

Tabelle 5.2.3 Aufwandsabschätzung in CLBs für den Fano–Decodern mit $\nu = 2$ bis 10 [MS93, S. 26].

- Die Decodierzeit ist von der Anzahl der Übertragungsfehler abhängig
- Aufgrund möglicher Übertragungsfehler muß die Decoderschaltung wesentlich höher getaktet sein, als es die ankommende Bit–Rate vorgibt

Wesentlich unkritischer im Fehlerfall verhält sich der **Stack–Decoder**. Dieser Decoder verwendet einen Kellerspeicher für l Pfade sowie deren Metriken. Der Kellerspeicher ist nach Metrikgröße sortiert. In jedem Schritt werden für die 2^k Verzweigungen des größten Metrikwertes die neuen Metriken berechnet und im Stapel einsortiert. Ein Decodierungsbeispiel ist z. B. in [Dor93, S. 156] dargestellt. Zum Ende der Decodierung befindet sich der optimale Pfad an oberster Stelle im Kellerspeicher.

Das Hauptproblem bei einer FPGA–Realisierung ist das Sortieren und Ordnen des Kellerspeichers. Dies ist in „Software“ relativ unproblematisch — hingegen in „Hardware“ sind insbesondere die durchzuführenden Vergleichsoperation recht aufwendig, siehe Tabelle 5.2.4.

Im Vergleich zum Fano–Decoder läßt sich festhalten: Es treten ähnliche Realisierungsprobleme auf. Der Stack–Decoder besitzt einen erhöhten Speicheraufwand

Funktionsgruppe	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$
Kellerspeicher	48	84	128	190
Vergleicher	21	40	56	72
Encoder	6	9	12	15
Multiplexer	6	8	10	12
Addierer	14	16	16	16
Steuerwerk	6	7	7	8
Summe	101	164	229	313

Tabelle 5.2.4 Aufwandsabschätzung in CLBs für den Stack–Decoder für $\nu = 2$ bis 5 [MS93, S. 28].

gegenüber dem Fano–Decoder — die nötigen Rechenoperationen sind aber wesentlich geringer. Der Stack–Decoder liegt bezüglich dem Realisierungsaufwand für eine FPGA–Implementierung zwischen Viterbi– und Fano–Decoder.

Aufgrund des möglichen Fehlverhaltens und der nicht vorhersehbaren Decodierzeit erscheinen sequentielle Decoder weniger für die Verwendung in einem Paging–Empfänger geeignet und werden daher im folgenden nicht weiter berücksichtig.

5.2.3.3 Algebraische Decoder

Algebraische Decoder werten, ebenso wie es in Abschnitt 5.1 bei Blockcodes erläutert wurde, das Fehlersyndrom aus. Im Decoder werden ebenso wie im Encoder (siehe Bild 5.2.11 und 5.2.12) aus den Informations–Bits die Parity–Bits berechnet. Diese werden mit den empfangenen Paritäten EXOR verknüpft und bilden nun das Syndrom. Ist das Syndrom ein Nullvektor, so trat kein Fehler auf — anderenfalls kann aufgrund des Syndrommusters auf die Fehlerstelle geschlossen werden. Im Gegensatz zu Blockcodes, die sich durch ein einziges Generatorpolynom beschreiben lassen, werden für Faltungscodes der Rate $R = k/n$ genau $k + 1$ Generatorpolynome benötigt. Diese Generatorpolynome werden für eine kompakte Darstellung zur $(n \times k)$–Generatorpolynom–Matrix zusammengefaßt. Für den systematischen Faltungscoder aus Abbildung 5.2.11 ergibt sich die Generatormatrix zu

$$\mathbf{G}(x) = 1 \qquad x^{21} + x^{20} + x^{19} + x^{17} + x^{16} + x^{13} + x^{11} + 1 \tag{5.2.3}$$

Es ergibt sich die für den systematischen Code zwingende Form für $G(x) = [I \vdots P(x)]$ aus Einheitsmatrix gefolgt von der Polynommatrix. Die Parity–Check–Matrix lautet somit $H(x) = [-P(x)^T \vdots I]$, da $G(x) \cdot H(x)^T = 0$ erfüllt sein muß. Damit berechnet sich der gesuchte Syndromvektor zu $S(x) = v(x) \cdot H(x)^T$, wobei $v(x)$ die empfangene Bit–Folge repräsentiert.

Die Syndrome entsprechen jeweils dem Produkt aus einem speziellen Fehlervektor und der Parity–Check–Matrix. Der **Syndrom–Decoder** erkennt nun anhand der berechneten Syndrome und einer (kleinen) Tabelle, ob dieses Fehlermuster korrigiert werden kann. Um die Tabelle klein zu halten, werden dabei nur Fehler korrigiert, die in der ersten Bit–Position auftreten. Kann ein Code mehrere Fehler korrigieren, so reicht es nicht aus, das Syndromregister zu löschen, sondern der Syndromwert muß vom Syndromregister subtrahiert werden, vergleiche Bild 5.2.12.

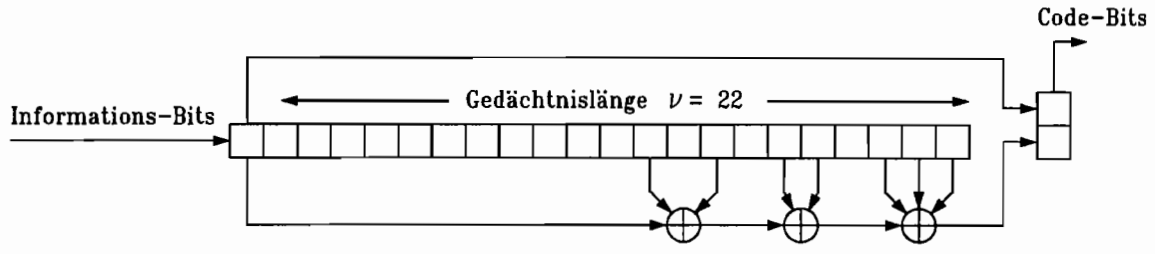


Bild 5.2.11 Systematischer $(44, 22)$–Encoder der Rate $R = 1/2$ und der Einflußlänge $\nu = 22$.

Nun wurde aber bereits im Abschnitt 5.1 gezeigt, daß mit FPGAs Tabellen mit mehr als 5 Eingangsvariablen schlecht realisierbar sind. Hier bieten eine spezielle Klasse von syndrom–decodierbaren Codes, die sogenannten „Majority–Codes", eine wesentliche Vereinfachung. Bei den „kanonisch selbstorthogonalen Codes" (CSOC) besitzen diejenigen J Spalten $\{A_k\}$ der Paritycheckmatrix, die als orthogonale Schätzwerte benutzt werden, lediglich in der ersten Zeile eine gemeinsame 1, siehe z. B. [CC88, S. 284]. Bei einem Auftreten eines Fehlers in der obersten Stelle sind somit mindestens $\lceil J/2 \rceil$ Einsen im Syndromregister vorhanden. Für die Decodierregel ergibt sich so:

$$e_0^i = \begin{cases} 1 & \text{für} \quad \sum_{k=1}^{J} A_k > \lceil J/2 \rceil \\ 0 & \text{sonst} \end{cases} \tag{5.2.4}$$

Nun wird unmittelbar die Bezeichnung „Majority–Code" einsichtig. Statt der aufwendigen Syndromtabelle wird lediglich ein Mehrheitsentscheider benötigt. Eine „Majority" decodierbare Klasse von Codes stellt Massey in [CC88, S. 289] unter der Bezeichnung „Trial and Error Codes" vor. Bei den zugehörigen Decodern werden nicht die Syndrome direkt ausgewertet, sondern die Syndrom–Bits werden so miteinander kombiniert, daß der resultierende Vektor orthogonal auf e_0^i steht. Diesem geringen zusätzlichen Hardware–Aufwand steht der Vorteil gegenüber, daß „Trial and Error Codes" im allgemeinen bei gleicher Einflußlänge etwas bessere Korrektureigenschaften als die CSOC–Codes besitzen. In Tabelle 5.2.3.3 sind einige „Trial and Error Codes" für die Rate 1/2 angegeben.

Tabelle 5.2.6 zeigt eine CLB–Aufwandsabschätzung für den Majority–Decoder für verschiedene J.

Zur Berechnung der Korrekturfähigkeit dieser Codes wird die vereinfachte Annahme getroffen, daß, ebenso bei Blockcodes (siehe Abschnitt 5.1.4), im Bereich der doppelten Einflußlänge der Code $\lfloor J/2 \rfloor$ Fehler korrigieren kann [Bla84, S. 440].

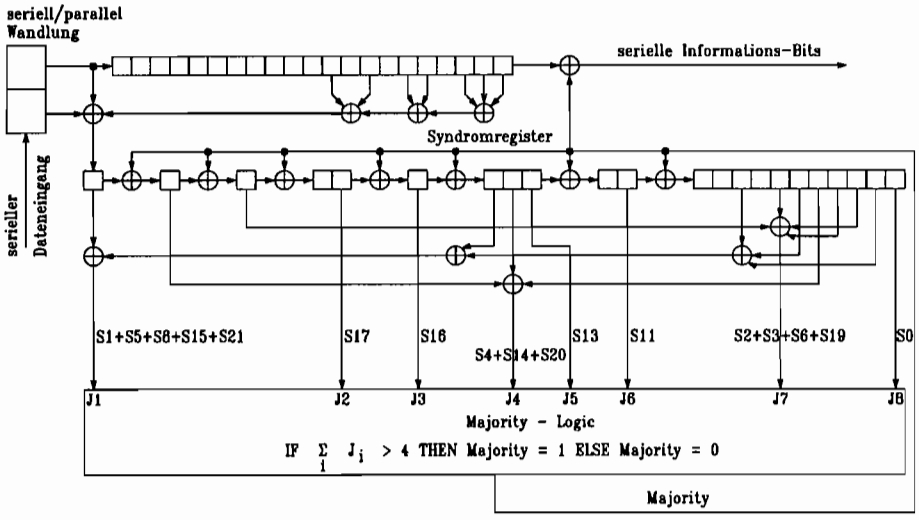


Bild 5.2.12 $J = 8$ Trial–and–Error Majority–Decoder.

J	t_{MD}	ν	Generatorpolynom	Orthogonalitätsgleichungen
2	1	2	$1+x$	s_0, s_1
4	2	6	$1+x^3+x^4+x^5$	s_0, s_3, s_4, s_1+s_5
6	3	12	$1+x^6+x^7+x^9+x^{10}+x^{11}$	$s_0, s_6, s_7, s_9, s_1+s_3+s_{10}$, $s_4+s_8+s_{11}$
8	4	22	$1+x^{11}+x^{13}+x^{16}+x^{17}+x^{19}$ $+x^{20}+x^{21}$	$s_0, s_{11}, s_{13}, s_{16}, s_{17}, s_2+s_3+s_6+$ $s_{19}, s_4+s_{14}+s_{20}, s_1+s_5+s_8+$ $s_{15}+s_{21}$
10	5	36	$1+x^{18}+x^{19}+x^{27}+x^{28}+x^{29}$ $+x^{30}+x^{32}+x^{33}+x^{35}$	$s_0, s_{18}, s_{19}, s_{27}, s_1+s_9+s_{28}, s_{10}+$ $s_{20}+s_{29}, s_{11}+s_{30}+s_{31}$, $s_{13}+s_{21}+s_{23}+s_{32}, s_{14}+$ $s_{33}+s_{34}, s_2+s_3+s_{16}+s_{24}+$ $s_{26}+s_{35}$

Tabelle 5.2.5 Liste einiger Majority decodierbaren „Trial and Error Codes" Faltungscodes [CC88, S. 406].

$$P(J) = \sum_{i=0}^{\lfloor J/2 \rfloor} \binom{2\nu}{i} p^i (1-p)^{2\nu-i} \tag{5.2.5}$$

Dabei ist das Fenster, in dem die $\lfloor J/2 \rfloor$ Fehler auftreten dürfen, „gleitend". Im Vergleich zu einem Blockcode gleicher Länge und gleicher Korrekturfähigkeiten ist dann dieser Faltungscode, geringfügig schlechter zu bewerten. Ab BFW kleiner 10^{-3} dürfte dies aber kaum noch einen Unterschied bedeuten. Anhand der durchgeführten Computer–Simulation über 10^6 Bit ergibt sich eine akzeptable Übereinstimmung mit den theoretisch berechneten Werten, siehe Bild 5.2.13.

Da die resultierende Einzelbitfehlerwahrscheinlichkeit berechnet werden soll, wird wiederum, wie im Abschnitt 5.1.4, die Blockfehlerwahrscheinlichkeit $P(J)$ mit der Blockfehlerwahrscheinlichkeit eines äquivalenten fehlerfreien Informationsblocks des (n, k) Codes

Funktionsgruppe	$J=4$	$J=6$	$J=8$	$J=10$
Register	6	12	22	36
EXOR–Gatter	2	4	7	11
Majority–Schaltung	1	5	7	15
Summe	9	22	36	62

Tabelle 5.2.6 Aufwandsabschätzung in CLBs für Majority–Decodern mit $J = 4$ bis 10.

gleichgesetzt. Für einen Code der Rate $R = 1/2$ ergibt sich so eine Blocklänge von $n - k = k$ und die zu berechnende Einzelbitfehlerwahrscheinlichkeit P_B ergibt sich aus

$$P(J) = P(0) = (1 - P_B)^k \tag{5.2.6}$$

$$\rightarrow P_B = 1 - e^{\ln(P(J))/k} \tag{5.2.7}$$

5.2.3.4 Abschließender Vergleich

Nachdem sich bereits in der Vorauswahl die sequentiellen Decoder als ungünstig für einen Paging–Empfänger erwiesen haben, bleiben Viterbi–Decoder und Majority–Decoder übrig. In Abbildung 5.2.13 sind Viterbi–Decoder und Majority–Decoder unterschiedlicher Einflußlänge gegenübergestellt. Es wurde dabei die bereits im Abschnitt 5.1.4 erläuterte Darstellung von Bitfehlerwahrscheinlichkeit mit und ohne Fehlerkorrektur gewählt.

Das Bild 5.2.13 in Verbindung mit den Tabellen 5.2.1 und 5.2.6 macht deutlich, daß bei gleichem CLB–Aufwand die Majority–Decoder aufgrund der wesentlich größeren Einflußlänge erheblich bessere Ergebnisse erzielen, als Viterbi–Decoder gleichen Aufwands.

Der für die Fehlerkorrekturschaltung im Paging–Decoder verfügbare Platz wurde mit ca. 50 CLB angegeben [Wat93], so daß letztlich der Majority–Decoder für den $J = 8$ „Trial–and–Error“ Code, entsprechend Bild 5.2.12, realisiert wurde.

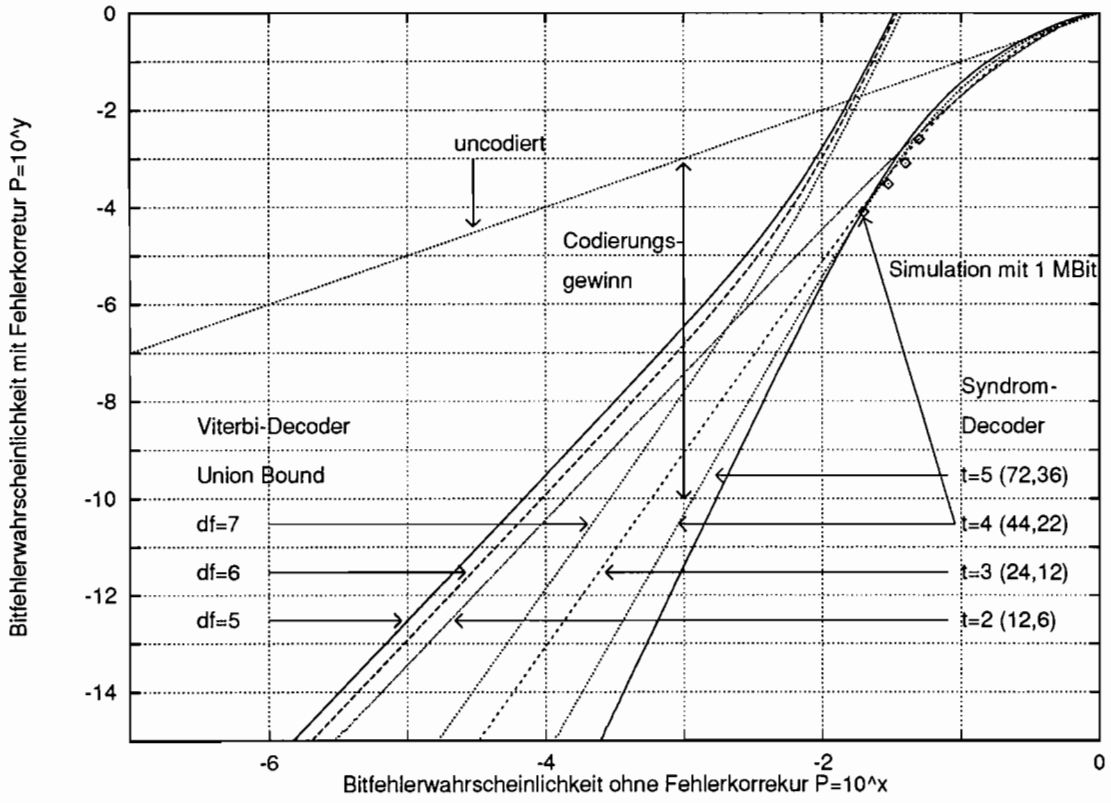


Bild 5.2.13 Vergleich von Viterbi– und Majority–Decoder.

5.2.4 Verschlüsselungsverfahren für FPGAs

Wie zu Beginn dieses Kapitels erläutert, wurden bei den hier vorgestellten Untersuchungen lediglich diejenigen Kryptographie–Algorithmen betrachtet und ausgewählt, die günstig in einem FPGA realisiert werden können. Günstig für den Implementierungsaufwand erscheinen hierbei Algorithmen, die ein rückgekoppeltes Schieberegister bei der Schlüsselerzeugung benutzen, und/oder mittels kleiner Tabellen (sogenannter Substitutionsboxen, S–Boxen) eine Verschlüsselung realisieren. Im folgenden wird kurz auf die prinzipiellen Unterschiede zwischen Strom–Chiffren und Block–Chiffren eingegangen, wobei Strom–Chiffren ebenso wie Faltungscode bei der Fehlerkorrektur für die kontinuierliche Datenbearbeitung bevorzugt werden.

5.2.4.1 Strom–Chiffren

Bei einer Strom–Chiffre wird die Nachricht, nachdem sie in einen seriellen Bit–Strom überführt wurde, mit einer Schlüsselfolge EXOR verknüpft, siehe Bild 5.2.14. Im Empfänger wird der Chiffre–Text erneut mit der gleichen Schlüsselfolge verknüpft. Bei der doppelt angewendeten EXOR–Verknüpfung entsteht wieder die Originalfolge:

$$K \oplus S \oplus S \;=\; K \oplus 0 = K \tag{5.2.8}$$

Rückgekoppelte Schieberegister maximaler Länge sind kryptographisch recht gute Annäherungen an einen idealen Schlüssel, da die von den Schieberegistern erzeugte Folge statisch günstige Eigenschaften besitzt, siehe [MS76], [LP73] oder [Sta73]. D. h. für einen kryptographischen Angriff sind die erzeugten Folgen schwierig zu analysieren. Ein vollständiger Zyklus dieser Folgen erfüllt die von Golomb formulierten drei Kriterien an eine Zufallsfolgen [BFS83, S. 188], [FR88, S. 132]:

1) Die Anzahl der Einsen und Nullen in einem Zyklus der Pseudozufallsfolge unterscheidet sich höchstens um eins.

2) Runs der Länge i (1–Folgen, 0–Folgen) haben an der Gesamtheit aller Runs in einem Zyklus den Anteil $1/2^i$.

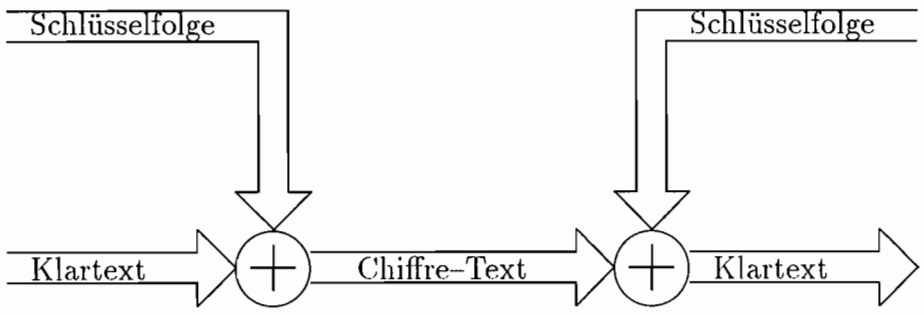


Bild 5.2.14 Prinzip einer Strom–Chiffre.

3) Die Autokorrelationsfunktion $C(\tau)$ ist konstant für $\tau \in [1, n-1]$. Dabei ist $C(\tau)$ folgendermaßen definiert:

$$C(\tau) = \frac{A(\tau) - D(\tau)}{n} = \frac{2A(\tau)}{n} - 1$$

Dabei ist n die Länge des Zyklus. $A(\tau)$ ist die Anzahl der übereinstimmenden Bit–Stellen zweier Zyklen, die um τ Stellen gegeneinander verschoben sind. $D(\tau)$ ist die Anzahl der nicht übereinstimmenden Bit–Stellen.

Eine weitere Verbesserung der kryptographischen Sicherheit wird durch die Verknüpfung mehrerer Schieberegisterausgänge zu einem Schlüssel erreicht. Hierbei gibt es eine Reihe von linearen und nichtlinearen Verknüpfungsmöglichkeiten [FR88, S. 150-173]. Günstig für den Realisierungsaufwand und die kryptographische Sicherheit haben sich die nichtlinearen Verknüpfungen mit Schwellenentscheid erwiesen. Bei einer Verknüpfung von 3 Schieberegister mit unterschiedlicher Länge L_1, L_2 und L_3 ergibt sich eine **lineare Komplexität**‡ zu $L_{\text{ges}} = L_1 \cdot L_2 + L_2 \cdot L_3 + L_1 \cdot L_3$.

Es soll nun eine konkrete Realisierung mit der Verknüpfung von 3 Schieberegistern durch einen Mehrheitsentscheider untersuchen werden. Abbildung 5.2.15(b) zeigt die zugehörige Schaltung. Als Rückkopplungspolynome wurden gewählt:

$$\begin{aligned} a_1(x) &= x^{33} + x^6 + x^4 + x + 1 \\ a_2(x) &= x^{29} + x^2 + 1 \\ a_3(x) &= x^{38} + x^6 + x^5 + x + 1 \end{aligned}$$

Alle Rückkopplungspolynome sind primitiv. Dies garantiert, daß jedes Schieberegister eine Folge maximaler Länge erzeugt. Für die linearen Komplexitäten gilt:

$$\begin{aligned} L_1 &= 33 \\ L_2 &= 29 \\ L_3 &= 38 \\ L_{\text{ges}} &= 33 \cdot 29 + 33 \cdot 38 + 29 \cdot 38 = 3313 \end{aligned}$$

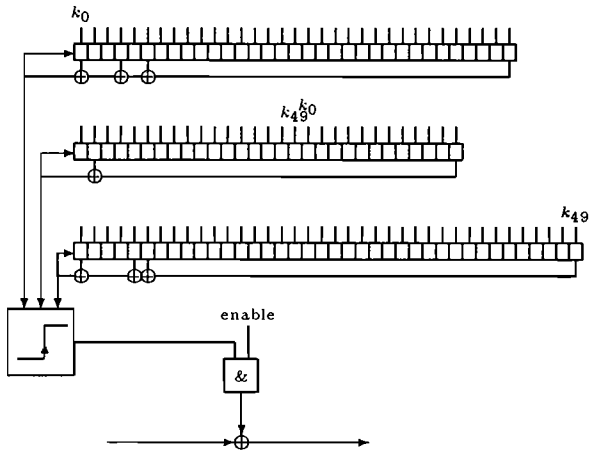
Da der Anfangszustand (der Schlüssel) bei jeder Verschlüsselung verloren geht, wird er in einem zusätzlichen Register gespeichert und zu Beginn in das Schieberegister geladen. Der Schlüssel ist 50 Bit breit und wird beim Laden der 100 Flip–Flops zweimal benutzt. Eine Hardware–Variante mit weniger Aufwand könnte den Schlüssel seriell in das Schieberegister laden. Er könnte dann von außen (μController, siehe Abschnitt 5.2.5) geladen werden. Dies würde jedoch den Ablauf komplizierter gestalten und soll an dieser Stelle nicht weiter betrachtet werden.

Für die Hardware–Realisierung sei nun noch eine Abschätzung für den CLB–Aufwand angegeben, siehe Bild 5.2.15(a).

‡ Unter linearer Komplexität verseht man hierbei die mit Hilfe z. B. des BerlekampMassey–Algorithmus bestimmte minimale äquivalente Schieberegisterlänge mit nur einem Schieberegister.

Funktionsgruppe	CLBs
50 Bit Schlüsselregister	25
100 Bit Schieberegister	50
Rückkopplungen	3
Mehrheitsentscheider	0,5
EXOR mit Nachricht	0,5
Summe	79

(a)



(b)

Bild 5.2.15 (a) Aufwand in CLB. (b) Realisierung des Strom–Chiffre mit 3 Schieberegistern.

5.2.4.2 Block–Chiffren

Ebenso wie rückgekoppelte Schieberegister können die sogenannten Substitutionsboxen mit bis zu 5 Variablen günstig mit den vorhanden LUT in einem CLB der Xilinx–FPGA realisiert werden, siehe Abschnitt 1.3. Im folgenden soll insbesondere betrachtet werden, wie der „Data Encription Standard“ (DES) für eine günstige FPGA–Implementierung modifiziert werden sollte.

Block–Chiffren arbeiten, wie der Name schon sagt, mit Nachrichtenblöcken. Diese Blöcke werden nach gewissen Algorithmen substituiert. Bild 5.2.16 zeigt einen typischen Zustandsautomaten für Block–Chiffren. Der zu substituierende Block wird zunächst in das Zustandsregister geladen und danach expandiert. Nach der Substitution in den S–Boxen werden noch Permutationen der einzelnen Bits durchgeführt. Dieser Zyklus kann mehrere Male wiederholt werden, wobei auch ein Wechsel des Schlüssels möglich ist.

Eine effiziente Sonderform der Block–Chiffren stellt die sogenannte Feistel–Chiffre dar, siehe Bild 5.2.17. Der Klartext wird in zwei Blöcke L_i und R_i zerlegt. R_i wird zusammen mit dem Schlüssel K_i über die Abbildung F_i verknüpft. Bei geeigneter Wahl der F_i wird der Code nach mehreren Schritten **vollständig**, d. h. jedes Ouput–Bit hängt von jedem Input–Bit ab. Zur Entschlüsselung wendet man sämtliche Operationen in umgekehrter Reihenfolge an.

Der DES ist eine iterative Anwendung der Feistel–Chiffre. Er ist in [FR88, S. 216-246] ausführlich beschrieben. Bei ihm werden die F_i folgendermaßen gebildet: Die 32 Bit des Registers R werden auf 48 Bit erweitert und zu einem Teilschlüssel K_i derselben Breite addiert. Das Ergebnis wird in 8 S–Boxen mit 6 Bit Breite substituiert und gleichzeitig wieder auf 32 Bit verkürzt. Eine abschließende Permutation bildet das Ergebnis der Funktion F_i. Die im DES verwendeten 6 Bit breiten S–Boxen sind für eine FPGA–Implementierung ungünstig. Daher wurde der DES leicht modifiziert, um mit 5 Bit breiten S–Boxen auszukommen. Allerdings mußten hierzu, siehe Tabelle 5.2.7, neue S–Boxen entworfen werden.

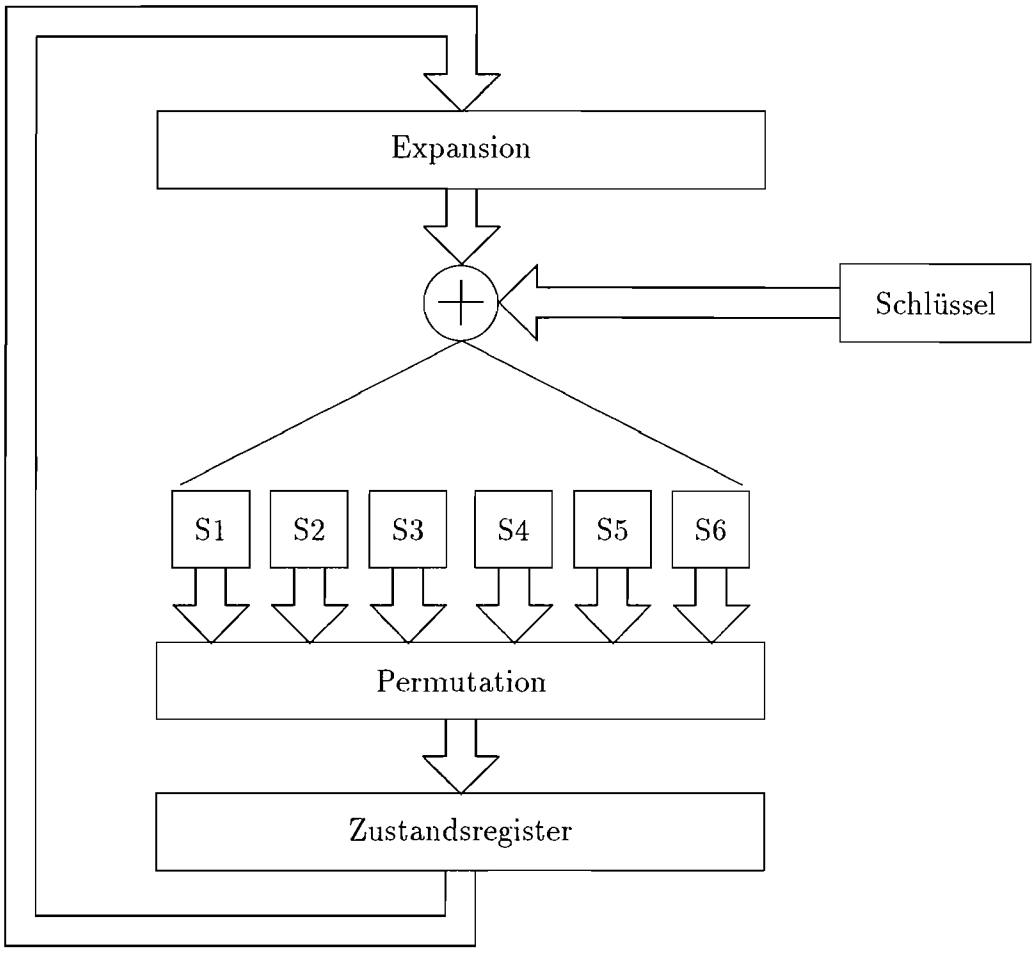


Bild 5.2.16 Zustandsautomat einer Block–Chiffre.

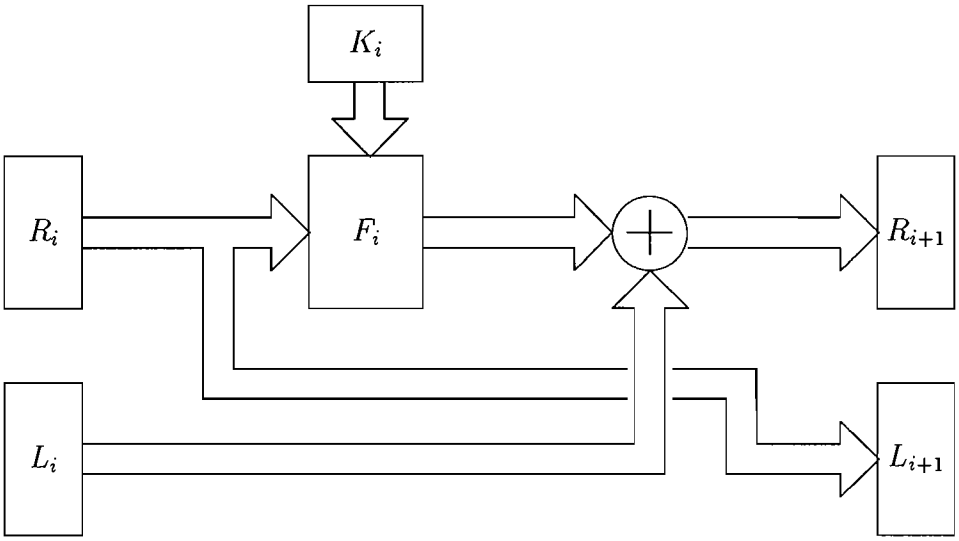


Bild 5.2.17 Prinzip der Feistel–Chiffre.

Die Betriebsarten ECB, CBC und CFB des DES kommen für das Pagingsystem nicht in Frage, da dabei die Nachricht direkt substituiert wird, siehe [FR88, S. 247-267] oder [Wat93, S. 18-20]. Aufgrund des Avalanche–Effekts [FR88, S. 196-203] würde ein einzelner Übertragungsfehler die gesamte Information des Blocks zerstören, da im Mittel 50% der Bits invertiert werden. Beim „Output Feedback Modus“ (OFB) hingegen wird

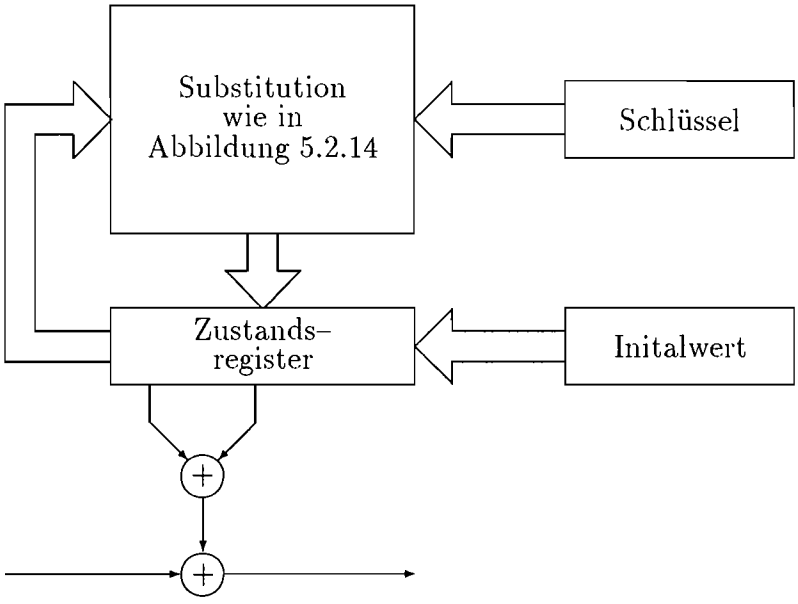


Bild 5.2.18 Block–Chiffre im OFB–Modus als Strom–Chiffre.

der Chiffrieralgorithmus als Pseudozufallsgenerator in einer Strom–Chiffre benutzt, siehe Bild 5.2.18. Der Ausgabeblock wird auf die Eingabe zurückgeführt, so daß eine pseudozufällige Folge von Ausgangsblöcken entsteht. Eine Auswahl von Bits des Ausgangs wird mit dem Klartext seriell EXOR–verknüpft. Im ersten Schritt besteht die Eingabe des Zufallsgenerators aus einen Initialisierungsvektor.

Im Abschnitt 5.2.4.3 werden sich die Vorteile des modifizierten DES im Bezug auf die kryptographische Sicherheit zeigen.

Das Zustandsregister habe hier eine Breite von 25 Bit. Eine Expansion findet hier nicht statt. Zur Substitution werden 5 S–Boxen verwendet, siehe Tabelle 5.2.7. Eine notwendige Bedingung für Block–Chiffren ist, daß die Substitution eindeutig umkehrbar ist. Da die Schaltung hier jedoch im OFB–Modus als Strom–Chiffre verwendet wird, muß diese Bedingung nicht unbedingt eingehalten werden. Trotzdem sind die gewählten S–Boxen alle eindeutig umkehrbar, so daß man sie auch für eine Block–Chiffre einsetzen könnte.

Eingang	Box 1	Box 2	Box 3	Box 4	Box 5
0	1E	F	14	19	6
1	13	1	1D	14	E
2	14	13	16	D	1A
3	1	1F	B	4	3
4	1A	19	5	1C	B
5	1B	1C	E	1A	1E
6	E	12	8	1E	0
7	B	11	F	1	2
8	D	8	4	C	1D
9	10	7	C	F	C
A	3	1B	1E	1B	18
B	0	0	13	1D	17
C	4	1A	10	5	1
D	6	C	1	15	15
E	A	1D	18	E	1B
F	17	2	17	13	9
10	19	B	1C	17	19
11	16	1E	A	9	A
12	7	18	1B	3	4
13	1C	D	3	10	14
14	1D	5	19	A	13
15	5	14	D	16	11
16	2	15	0	12	10
17	1F	9	2	1F	12
18	F	3	15	B	5
19	11	10	6	2	F
1A	C	6	7	6	8
1B	18	17	12	18	16
1C	9	4	1F	11	1C
1D	15	16	1A	8	7
1E	8	E	9	7	D
1F	12	A	11	0	1F

Tabelle 5.2.7 Die 5 Substitutionsboxen.

Box 1					Box 2					Box 3				
20	12	20	20	20	20	16	20	12	20	20	12	16	16	16
12	20	12	16	16	20	20	20	16	16	16	20	16	16	16
12	16	16	12	8	12	20	20	16	8	16	16	20	12	12
16	16	20	12	16	16	24	12	16	12	16	8	12	16	20
20	16	20	12	12	16	20	16	20	20	20	12	12	20	12

Box 4					Box 5				
20	16	20	20	16	12	20	8	12	20
12	16	12	16	20	20	12	16	24	20
20	16	16	20	16	16	12	12	20	16
20	16	16	20	24	16	20	16	20	12
16	12	28	20	16	12	16	16	12	24

Tabelle 5.2.8 Abhängigkeitsmatrizen der S–Boxen (Idealwert 16).

Ein wichtiges Instrument zur Beurteilung der kryptographischen Sicherheit der S–Boxen stellt die sogenannte Abhängigkeitsmatrix dar. Sie enthält für jede Kombination von Eingangs– und Ausgangs–Bit die Wahrscheinlichkeit, daß sich das Ausgangs–Bit ändert, wenn sich das Eingangs–Bit ändert. Im Idealfall beträgt die Wahrscheinlichkeit 1/2 (Avalanche–Effekt, [FR88][S. 196-203]). Tabelle 5.2.8 zeigt die Abhängigkeitsmatrizen der S–Boxen. Statt der Wahrscheinlichkeiten wird die absolute Anzahl der Änderungen eingetragen. Da es insgesamt $2^5 = 32$ mögliche Eingangsvektoren für jede S–Box gibt, beträgt der Idealwert 16. Zur Erzeugung der S–Boxen wurde ein Zufallsgenerator eingesetzt. Die Iteration wurde dann abgebrochen, als die Abhängigkeitsmatrizen brauchbar erschienen. Daß einige Einträge trotzdem stark vom Idealwert abweichen, ist eine Folge davon, daß die Forderung nach der eindeutigen Umkehrbarkeit die Auswahl einschränkt.

Tabelle 5.2.9 zeigt die Permutationsvorschrift. Man beachte, daß sich alle Ausgangs–Bits einer S–Box im nächsten Zyklus auf alle anderen S–Boxen verteilen. Bei der Wahl der Permutation ist insbesondere darauf zu achten, daß die Bits von Schritt zu Schritt in andere S–Boxen hineingehen. Sonst könnte das Zustandsregister in Teile

von Bit Nr.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
nach Bit Nr.	20	4	5	10	15	21	0	6	11	16	22	1	7	12	17
von Bit Nr.	15	16	17	18	19	20	21	22	23	24					
nach Bit Nr.	23	2	8	13	18	24	3	9	14	19					

Tabelle 5.2.9 Permutationstabelle.

zerfallen, die unabhängig voneinander Zyklen verschiedener Länge durchlaufen, und nicht alle dieser Zyklen wären für die Schlüsselfolge relevant.

Der Schlüssel wird von einem Zyklus zum nächsten nicht verändert. Zur Verknüpfung mit dem Klartext werden nur 2 Bits des Zustandsregisters verwendet. Da am Anfang des Verschlüsselungsvorgangs das Zustandsregister mit einem Initialisierungsvektor geladen werden muß, ist für den Initialisierungsvektor ein eigenes Register vorgesehen. Als Abschätzung für den Hardware–Aufwand (Anzahl der CLBs) ergibt sich:

Funktionsgruppe	CLBs
25 Bit Schlüsselregister	12,5
25 Bit Addition	12,5
25 Bit Zustandsregister	12,5
5 S–Boxen 5:5	25
Permutation	0
25 Bit Initialisierungsvektor	12,5
Multiplexer zur Auswahl: Initialisierungsvektor/S–Box	12,5
Verknüpfung mit der Nachricht	1
Summe	87,5

5.2.4.3 Vergleich der Verfahren bezüglich der kryptographischen Sicherheit

In [Wat93] wurden sechs verschiedene Vergleichskriterien entworfen und auf die zwei FPGA–Implementierungen (3SR, DES–OFB) angewendet. Von diesen Ergebnissen sollen hier die zwei aussagekräftigsten Kriterien 4 und 5 aufgeführt werden.

4) Wenn sich zwei Schlüssel in genau einem Bit unterscheiden, so werden sich auch die erzeugten Ausgangsfolgen unterscheiden. Dabei sollte sich im Mittel die Hälfte aller Bits der Ausgangsfolge ändern (Avalanche–Effekt).

5) Wie bei Kriterium 4 wird auch hier untersucht, wie sich eine Änderung in einem Bit des Schlüssels auf die Ausgangsfolge auswirkt. Hier wird jedoch für jede Stelle der Ausgangsfolge über alle möglichen Schlüssel–Bit–Änderungen aufsummiert. Nimmt man vereinfachend an, das alle Schlüsseländerungen gleich wahrscheinlich sind, so sollte sich jedes Ausgangs–Bit mit der Wahrscheinlichkeit 1/2 ändern (striktes Avalanche–Kriterium).

Für die Kriterien 4 und 5 wird mit einem Schlüssel eine Master–Schlüsselfolge der Länge 64 Bit erzeugt, siehe Bild 5.2.14. Der Klartext ist dabei beliebig. Anschließend wird der zufällig ausgewählte Schlüssel in jedem seiner 50 Bits variiert und wieder zur Erzeugung einer Schlüssel–Folge verwendet. Die Unterschiede zur Master–Folge werden für jede Variation über alle 64 Bit–Stellen aufsummiert (Kriterium 4). Außerdem werden für jede Bit–Stelle die Änderungen über alle Variationen aufsummiert (Kriterium 5). Diese Untersuchung wird mit 100 zufällig ausgewählten Schlüsseln durchgeführt und die Ergebnisse aufsummiert. Daraus ergibt sich für Kriterium 4 ein Idealwert von $64 \cdot 0,5 \cdot 100 = 3200$ und für Kriterium 5 von $50 \cdot 0,5 \cdot 100 = 2500$. Die Abbildungen 5.2.19 und 5.2.20 zeigen die Testergebnisse.

Die Ergebnisse nach Kriterium 4 zeigen deutlich, daß die DES–OFB–Version viel empfindlicher auf Schlüsseländerungen reagiert als die SR–Version. Darüberhinaus kommt die OFB–Version dem Idealwert sehr viel näher als die SR–Version.

Betrachtet man die Auswertung zu Kriterium 5, so wird es offensichtlich, daß sich Änderungen einzelner Schlüssel–Bits bei der SR–Version erst sehr spät (hier nach 32 Schritten) auf die Ausgangsfolge auswirken. Bei der DES–OFB–Version dagegen wirken sie sich sehr schnell aus. Nur die ersten 4 Ausgangs–Bits weichen vom Idealverhalten merklich ab. Alle anderen Bits werden sehr gleichmäßig beeinflußt. Bei der SR–Version sind auch nach dem 32. Schritt noch größere Schwankungen erkennbar.

Im folgenden wird erklärt, warum die DES–OFB–Variante kryptographisch bessere Eigenschaften besitzt als die 3SR–Variante:

Die Schieberegister sind vor allem deshalb so einfach zu beschreiben, weil in jedem Schritt nur 1 Bit durch das Rückkopplungspolynom berechnet wird, während alle anderen Bits lediglich geschoben werden. Die maximale Länge, nachdem eine Wiederholung des Zyklus für die 3 Schieberegister auftritt, ergibt sich mit $L = \sum L_i$ zu:

$$\prod_{i=1}^{3}(2^{L_i} - 1) \approx 2^{\sum L_i} = 2^L \tag{5.2.9}$$

Für die gewählte Registerlänge ergibt sich mit $L = 33 + 29 + 38 = 100$ eine Zykluslänge von 2^{100}. Auch bei Verknüpfung von 3 Schieberegistern bleibt die Anzahl der Zustandsübergänge beschränkt. Nimmt man vereinfachend an, daß man in jedem Zustandsübergang 3 neue Bits frei wählen kann, so ergeben sich 8 mögliche Zustandsübergänge und als maximale Anzahl der möglichen Zyklen maximaler Länge ergibt sich $Z_1 = 8^{2^L}$. Verwendet man in den 3 Teil–Schieberegistern primitive Rückkopplungspolynome, so kommt, bedingt durch die beschränkte Anzahl primitiver Polynome, außerdem

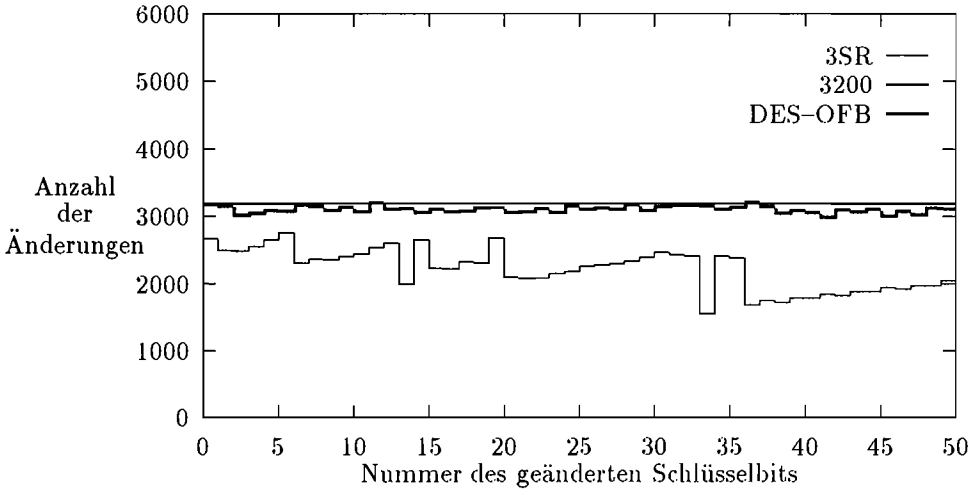


Bild 5.2.19 Untersuchung von Kriterium 4.

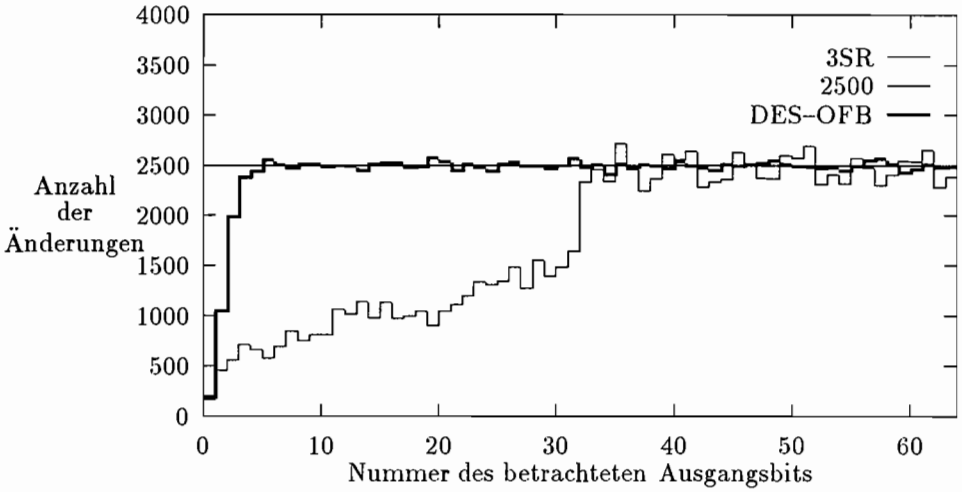


Bild 5.2.20 Untersuchung von Kriterium 5.

nur eine geringe Anzahl von Zyklen in Betracht. Der wesentliche Teil des kryptographischen Schutzes liegt also in der (geheimen) Wahl des Startzustands der Schieberegister. Die Einfachheit der Schieberegister läßt vermuten, daß die kryptographische Sicherheit dieser Chiffren nicht optimal ist.

Könnte man in jedem Zustand einen Folgezustand derart frei wählen, daß sich kein Zyklus kürzer als 2^L ergibt, so könnte man rund $(2^L)!$ verschiedene Zyklen maximaler Länge erzeugen. Mit Hilfe der „Stirlingschen Formel" [BS84, S. 103] läßt sich die Fakultät für große Argumente folgendermaßen abschätzen:

$$\ln(2^L!) \;=\; (2^L + 0,5) \cdot L \cdot \ln(2) - 2^L + 0,5\ln(2\pi) \tag{5.2.10}$$

Wird die Anzahl der möglichen Zyklen maximaler Länge der zwei Verfahren verglichen, so ergibt sich

$$Z_1 = e^{3\cdot 2^L \ln(2)} = e^{2,6\cdot 10^{30}} \;\ll\; Z_2 = e^{(2^L+0,5)\cdot L\cdot\ln(2) - 2^L + 0,5\ln(2\pi)} = e^{8,7\cdot 10^{31}} \tag{5.2.11}$$

Aus dem Quotient Z_2/Z_1 ergibt sich, daß es beim zweiten Verfahren um einen Faktor $e^{8,4\cdot 10^{30}}$ mehr Möglichkeiten bei der freien Wahl der Übergänge gibt. Die Sicherheit ist somit wesentlich besser und ein Kryptoanalytiker, der einen Teil eines solchen Zyklus kennt, hat erheblich mehr Schwierigkeiten, daraus auf die allgemeine Berechnungsvorschrift für Folgezustände (und damit auf den Schlüssel) zu schließen. Ein solches Verknüpfungsnetz ist jedoch nur mit großem Aufwand realisierbar, z. B. mit einer Tabelle zur Bestimmung der Folgezustände.

Bei der DES–OFB–Schaltung entsprechend Abbildung 5.2.16 entsteht zwar keine vollständige Freiheit in der Wahl des Folgezustands, dennoch ist die Freiheit wesentlich größer als bei der 3SR–Variante. Die Komplexität der Schlüsselfolge läßt sich weiter steigern, indem man von einem Verschlüsselungsvorgang zum nächsten mehrere Schritte durchführt.

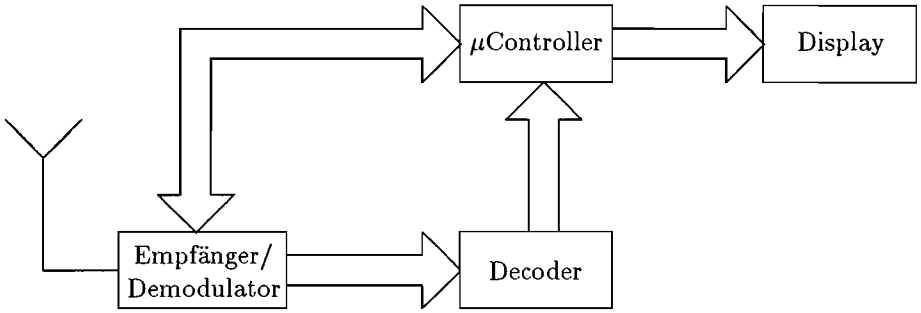


Bild 5.2.21 Blockdiagramm des Empfängers.

5.2.5 Hardware–Struktur des gesamten Decoders

Der Empfänger läßt sich gemäß Abbildung 5.2.21 zunächst grob in vier Blöcke aufteilen:

- den Empfangsteil mit Demodulator
- den Decoder, der in einem FPGA implementiert wurde
- einen µController zur allgemeinen Steuerung des Empfängers
- ein Display, auf dem der µController die empfangenen Nachrichten anzeigt

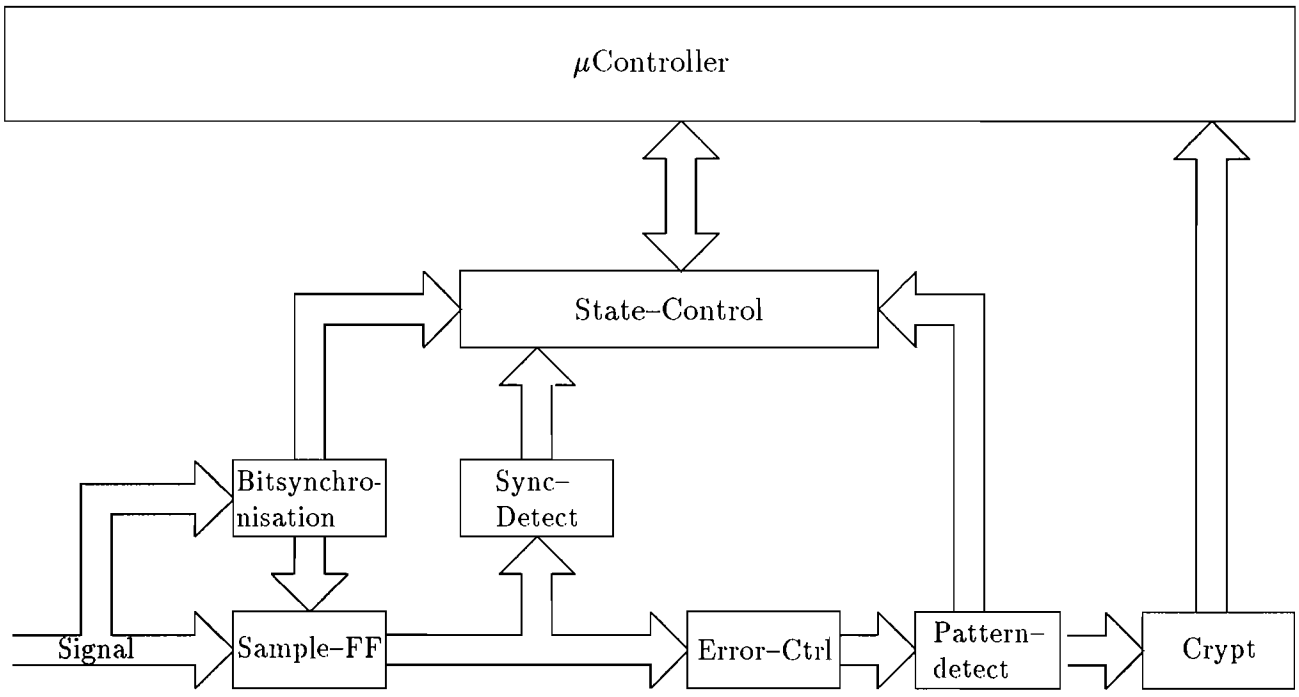


Bild 5.2.22 Blockdiagramm des FPGAs.

5.2.5.1 Blockstruktur des Decoder–FPGAs

Das Blockschaltbild des Decoders ist in Bild 5.2.22 dargestellt und soll im folgenden stichwortartig erläutert werden. Das FPGA tastet das vom Empfangsteil gelieferte Signal mit einem D–Flipflop ab. Dazu beobachtet der Block „Synchronisation" das Signal und steuert den zeitlichen Ablauf der Abtastung und des restlichen FPGAs auf Bit–Ebene.

Der Block „Sync–Detect" dient dazu, das Auftreten der zur Synchronisation verwendeten Barker–Folge anzuzeigen. In „Error–Ctrl" findet die Fehlerkorrektur statt. „Patterndetect" zeigt das Auftreten der Empfängeradresse und des Zeit–Tokens an. „Crypt" entschlüsselt die Daten, bevor sie zum μController übertragen werden. State–Control steuert die Abläufe im FPGA und die Signale zum μController, siehe Bild 5.2.23.

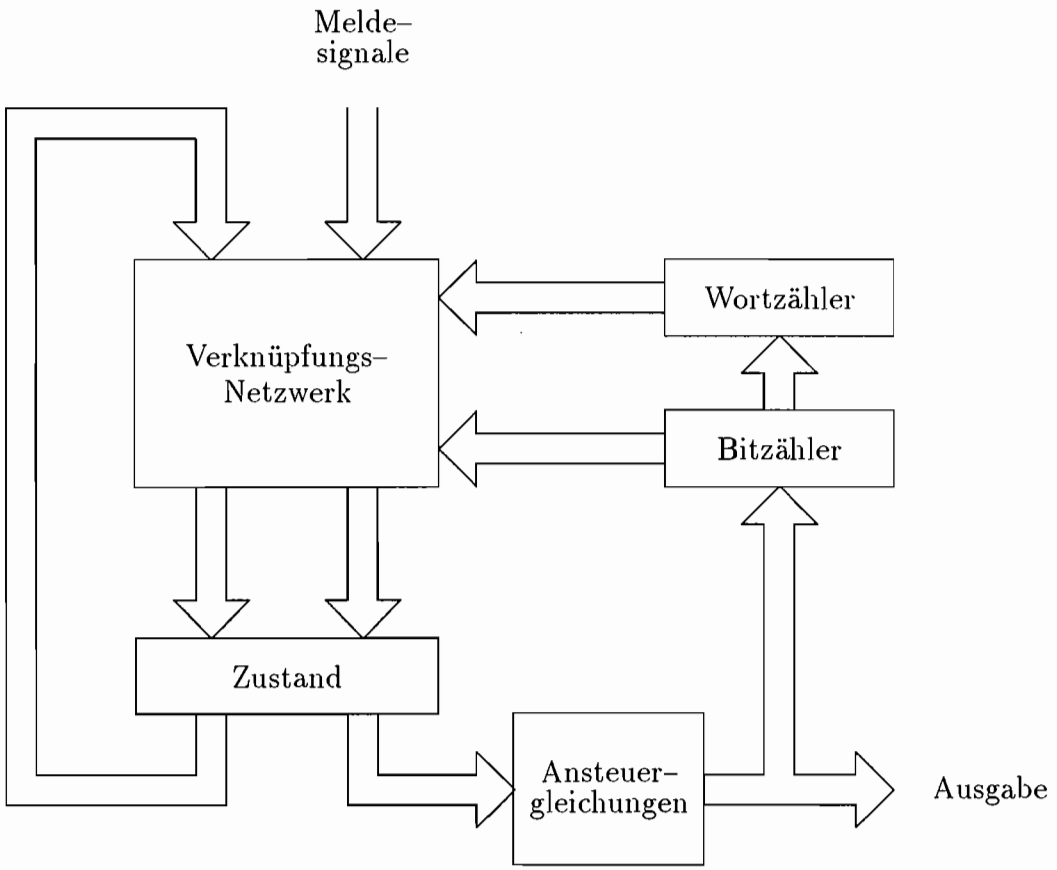


Bild 5.2.23 Struktur der Ablaufsteuerung.

Bei der Schaltung handelt es sich um ein Zweiphasensystem. Die Datenübernahme sämtlicher Datenregister findet mitten zwischen den Abtastzeitpunkten statt, die durch die Bit–Synchronisation vorgegeben sind. Die Zustandsübergänge der Ablaufsteuerung dagegen finden gleichzeitig mit der Abtastung statt, so daß der neue Zustand die Ansteuerung für die nächste Datenübernahme bestimmt.

Eine Abschätzung für die Verteilung der CLBs auf die einzelnen Funktionsblöcke wird in der Tabelle 5.2.10 dargestellt (vgl. hierzu auch Abbildung 5.2.22). Diese

Funktionsgruppe		CLBs
Bit–Synchronisation		4
Sample–Flipflop		0,5
Sync–Detect		3,5
Error–Ctrl		36
Crypt (OFB–Version)		87,5
Patterndetect	16 Bit Register Zeit–Token	8
	16 Bit Adreßregister	8
	16 Bit Schieberegister	8
	2 x 16 Bit Komparator	16
State Control (siehe Bild 5.2.23)	Verknüpfungsnetzwerk	50
	5 Bit Zustandsregister	2,5
	Ansteuernetzwerk	5
	Wortzähler	10
	Bit–Zähler	12
Summe		250

Tabelle 5.2.10 Aufwandabschätzung für die Realisierung des Decoder mit FPGA.

Decoder–Schaltung konnte in einem XC3090–FPGA mit 320 CLBs relativ problemlos plaziert und verdrahtet werden.

Kapitel 6

Zusammenfassung

In der vorliegenden Arbeit werden komplexe Algorithmen der Empfängertechnik auf deren Implementierbarkeit in PGA–Technik hin untersucht. Dazu gehören die Bereiche Frequenzerzeugung, Selektion, A/D–Wandlung, Demodulation und Codierung für eine Langwellenübertragung.

Im Bereich der **Frequenzerzeugung** zeigte sich, daß für einen in weiten Bereichen durchstimmbaren Frequenzsynthesizer bei geringem Leistungsverbrauch und kleinem Phasen– bzw. Frequenzfehler weder rein digitale noch rein analoge Verfahren geeignet sind. Hier wurde ein neues hybrides Verfahren gefunden [MB92b], das sich durch geringen Leistungsverbrauch, kleine Schaltungskomplexität und eine außerordentlich kurze Einschwingzeit auszeichnet. Die meßtechnische Verifikation der PGA–Realisierung konnte die überlegenen Eigenschaften des neuen Verfahrens bestätigen [MBW94b], [MBW94a].

Im Bereich der **Selektion und A/D–Wandlung** wurde ein speziell auf Langwellenabtastempfänger hin ausgelegter Entwurfsalgorithmus für Bandpaß–$\Sigma\Delta$–A/D–Wandler entwickelt, der eine Unterdrückung niederfrequenter Störungen und ein verbessertes Aliasing–Verhalten ohne zusätzlichen Schaltungsaufwand ermöglicht [MB92a], [MBH94a]. Bei der Selektion wurde eine neue Klasse von Frequenzabtastfiltern zur digitalen Filterung von Bandpaßsignalen entwickelt [MB93a], die eine wesentliche Aufwandsreduktion von über 90% erreicht, und dadurch eine PGA–Realisierung erst ermöglicht. Mittels einer parallelen DSP/FPGA–Struktur konnte gezeigt werden, daß dieses Filter für beliebige Eingangssignale stabil arbeitet (Force Response Stability).

Die Untersuchung der verschiedenen **Demodulatoren** zeigte, daß das neu entwickelte, inzwischen patentierte Verfahren der Hilbert–Abtastung mit CORDIC–Demodulation eine wesentliche Leistungssteigerung gegenüber der Quadraturabtastung und auch gegenüber der algebraischen Demodulation erlaubt [MB91], [MBH92], [MB93b] und [HMB94]. Liegen FM–modulierte Signale mit sehr kleinem Modulationsindex vor, wie z. B. bei der WeFAX–LW–Übertragung, und ist eine sehr gute Bit–Synchronisation vorhanden, so ist es eine neue Erkenntnis, daß bei der fortlaufenden inkohärenten I&D–Demodulation und geeigneter Filterung mit nur zwei CORDIC–Wandlungen $X, Y \rightarrow R, \theta$ pro Bit–Intervall sich eine bessere Bitfehlerwahrscheinlichkeit ergibt als bei allen anderen Demodulatoren. Außerdem konnte gezeigt werden, daß das erstmals auf CORDIC–gewandelte Phasenwerte angewendete LMS–Verfahren mittels „Knickgeraden“ für eine *gedächtnisbehaftete* schmalbandige FM–Demodulation bei geringfügig erhöhtem Aufwand wesentlich kleinere Bitfehlerwahrscheinlichkeiten erreicht, als bei Berücksichti-

gung nur des aktuellen Bit–Intervalls. Gegenüber bisherigen Veröffentlichungen ist es ebenso eine neue Erkenntnis, daß die CORDIC–Phasenwerte θ bei Koordinatenwandlungen $X, Y \rightarrow R, \theta$ nur asymptotisch die Betragsauflösung erreichen [MB93b], [MB+94]. Der abschließende Vergleich der verschiedenen Demodulatoren ergab für die realisierten kohärenten Empfänger, daß bei erhöhtem Aufwand für die Selektion in der analogen Eingangsstufe die eigentliche Demodulatorschaltung mit weniger Silizium–Fläche in einem PGA realisiert werden kann als die inkohärenten Demodulatoren. Für die realisierten kohärenten und inkohärenten Empfänger ergab sich eine vergleichbare Empfangsqualität [MB89].

In Bezug auf **zukünftige Codierschemata für die Langwellendatenübertragung** konnte gezeigt werden, wie das Funkuhr–Codierschema zum bisherigen Format kompatibel ergänzt werden kann, so daß bei einem angenommenen maximalen Fehlverhalten des Empfängers ein Fehlverhalten bei Vorhandensein der vorgeschlagenen Fehlerkorrektur nur einmal in drei Jahren auftritt. Dabei ist besonders interessant, daß die Fehlerkorrektur „nichts kostet", da die für die Fehlerkorrektur vorgesehenen Bit–Positionen im Übertragungsprotokoll z. Z. ungenutzt übertragen werden. Die FPGA–Implementierung der Fehlerkorrekturschaltung fand in einem der kleinsten FPGAs Platz [GHMB93].

Beim Entwurf eines Paging–Systems konnte von den Vorteilen der LW–Technik bei unidirektionaler Datenübertragung Gebrauch gemacht werden [MBH94b]. Neben den günstigen Ausstrahlungs– und Durchdringungseigenschaften der Langwelle erscheint gegenüber höherfrequenten Systemen vor allem die Tatsache vorteilhaft, daß nur eine einzige Sendeantenne benötigt wird. Hierfür wurde ein vollständiges Protokoll zur Ansteuerung von bis zu $9 \cdot 10^6$ Empfängern entworfen und die zugehörige FPGA–Schaltung einschließlich der Fehlerkorrektur durch einen Faltungsdecoder und eine Verschlüsselung der Nachrichten mit einem modifizierten DES–Algorithmus realisiert. Es konnte gezeigt werden, daß selbst sehr komplexe Algorithmen zur Fehlerkorrektur und Verschlüsselung vorteilhaft mit einem PGA realisiert werden können.

Zusammenfassend bleibt festzuhalten, daß es mit den vorgeschlagenen Algorithmen möglich ist, auch in PGA–Technik — schnell und kostengünstig — leistungsfähige, flexible und universelle Abtastempfänger zu realisieren.

Literaturverzeichnis

[AAS86] J. Anderson, T. Aulin und C.-E. Sundberg. *Digital Phase Modulation.* Plenum Press, 1986.

[ACM82] G. Ascheid, Y. Chen und H. Meyr. *Synchronisation, Demodulation und Dekodierung bei bandbreiteneffizienter Übertragung. NTZ Archiv*, Seiten 355–363, 1982.

[Ake78] B. Akers. *Binary Decusion Diagrams. IEEE Transactions on Computers*, Seiten 509–516, Juni 1978.

[AS83] B. P. Agrawal und K. Shenoi. *Design Methodology for $\Sigma\Delta M$. IEEE Transactions on Communications*, Seiten 360–369, März 1983.

[Aue88] E. Auer. *Determination of all existing Limit Cycles in High-Order Cascade- and Parallel-Structures of Recursive Digital Filters. ISCAS*, Seiten 2739–2742, August 1988.

[Bel89] U. Bellmann. *Meßplatz zum Prüfen und Entwickeln von Funkuhrempfängern.* Dissertationsschrift, Institut für Datentechnik, THD, 1989.

[Ber85] R. Bermbach. *Neue Funkuhrkonzepte durch enge verflechtung von Empfänger und Mikrocomputer.* Dissertationsschrift, Institut für Datentechnik, THD, 1985.

[Bes87] R. Best. *Theorie und Anwendung des Phase–locked Loops.* AT Verlag, 1987.

[BF77] C. W. Barnes und A. T. Fam. *Minimum Norm Recursive Digital Filters that Are Free of Overflow Limit Cycles. IEEE Transactions on Circuits and Systems*, Seiten 569–574, Oktober 1977.

[BFS83] P. Bratley, B. Fox und L. Schrage. *A Guide to Simulation*, Seiten 186–190. Springer–Lehrbuch, 1983.

[BL92] S. Beyer und G. Lipperer. *Stromsparendes Oszillatorkonzept für GSM-Anwendung. Siemens–Components*, Seiten 23–25, Januar 1992.

[Bla] Blaupunkt. *Costas Loop für RDS.* Deutsches Patent DE 3510562A1

[Bla84] R. E. Blahut. *Theory and practice of error control codes.* Addison–Wesley, 1984.

[BM87] J. Bergmann und J. Müller. *Entwurf und Aufbau eines Sigma–Delta–Modulators zur A/D–Wandlung.* Studienarbeit ST 329, IDT, 1987.

[Bol84] A. G. Bolton. *A Two's Complement Overflow Limit Cycle Free Digital Filter Structure. IEEE CAS Transactions Brief*, Seite 1045, Dezember 1984.

[Bro90] D. P. Brown. *Overflow Stability of N-Order State Space Digital Filters. Journal of the Franklin Institute*, Seiten 877–891, Vol. 327, 1990.

[Bry86] R. Bryant. *Graph–Based Algorithms for Boolean Function Manipulation. IEEE Transactions on Computers*, Seiten 677–691, August 1986.

[BS84] I. N. Bronstein und K. A. Semendjajew. *Taschenbuch der Mathematik.* Verlag Harri Deutsch, 1984.

[Bur93] D. Bursky. *Denser, Faster FPGAs vie for Gate–Array Applications. Elektronic Design*, Seiten 55–75, Mai 1993.

[Can85] J. C. Candy. *A Use of Double Integration in Sigma Delta Modulation. IEEE Transactions on Communications*, Seiten 249–258, März 1985.

[CC88] G. Clark und J. Cain. *Error–Correction Coding for Digital Communications.* Plenum Press, 1988.

[Cho87] H. Choi. *Effect of Gain and Phase Imbalance on the Performance of Lock Detector of Costas Loop. IEEE International Conference on Communications, Seattle*, Seiten 218–222, Juni 1987.

[CK74] T. Claasen und L. Kristiansson. *Improvement of Overflow Behaviour of 2nd-Order Digital Filters by Means of Error Feedback. Electronics Letters*, Seiten 240–241, Juni 1974.

[CMP75] T. Claasen, W. Mecklenbräuker und J. Peek. *Frequency Domain Criteria for the Absence of Zero-Input Limit Cycles in Nonlinear Discrete-Time Systems, with Applications to Digital Filters. IEEE Transactions on Circuits and Systems*, Seiten 232–246, März 1975.

[CMP76a] T. Claasen, W. Mecklenbäuker und J. Peek. *Effects of Quantization and Overflow in Recursive Digital Filters. IEEE Transactions on Acoustics, Speech and Signal Processing*, Seiten 517–529, Dezember 1976.

[CMP76b] T. Claasen, W. Mecklenbäuker und J. Peek. *On the Stability of the Forced Response of Digital Filters with Overflow Nonlinearities. IEEE Transactions on Acoustics, Speech and Signal Processing*, Seiten 692–696, Dezember 1976.

[Cos56] J. Costas. *Synchronous Communications. Proceedings of the IRE*, Seiten 1713–8, 44 / 1956.

[CR83] R. E. Crochiere und L. R. Rabiner. *Multirate Digital Signal Processing.* Prentice Hall, 1983.

[CS93] Inc. Crosspoint Solutions. *Advance Information.* CP20K Series, Januar 1993.

[DA85] P.R. Diniz und A. Antoniou. *Low-Sensitivity Digital-Filter Structures which are Amenable to Error-Spectrum Shaping. IEEE Transactions on Circuits and Systems*, Seiten 1000–1007, Oktober 1985.

[Dav82] F. Davarian. *Phase Error Statistics in Costas Loops Due to Channel Distortion and Noise. IEEE Transactions on Communications*, Seiten 2265–70, Oktober 1982.

[DD94] F. Dresig und A. Ditzinger. *Evolution – Logiksynthese für komplexe PGA. Elektronik Praxis*, Seiten 32–39, April 1994.

[Dor93] B. Dorsch. *Codierung zur Fehlerkorrektur.* Vorlesungs–Skript, SS 1993.

[DP89] P. Dillian und I. Phillips. *Elektrisch rekonfigurierbare Arrays: die „Sofort–ASICs". Elektronik*, Heft 21, 1989. Sonderdruck.

[dR93] R. del Rio. *Synthese Boolescher Funktionen durch Field–Programmable Gate Arrays.* Diplomarbeit dt, Uni FFm, 1993. FB Informatik.

[Dre91] H.-J. Dreßler. *Interpolative A/D-Conversion of Bandpass-Signals. Archiv der elektrischen Übertragung (AEÜ)*, Seiten 30–37, Band 45, 1991.

[Dru89] S. Drude. *Zwei ICs für einen Pager. Funkschau*, Seiten 69–76, 26 / 1989.

[E+92] R. Eckl et al. *Schwerpunkt: Wandler. Elektronik Industrie*, Seiten 30–58, April 1992.

[EB91] G. Edwards und S.W. Best. *Direkt-Synthesizer – Überblick und Applikation. Elektronik Industrie*, November 1991.

[EMR87] G. Engeln-Müllges und F. Reuter. *Numerische Mathematik für Ingenieure.* BI Wissenschaftsverlag, 1987.

[EMT69] P. Ebert, J. Mazo und M. Taylor. *Overflow Oscillations in Digital Filters. The Bell System Technical Journal*, Seiten 2999–3020, November 1969.

[ESB93] Eggert, R. Saupe und M. Biedermann. *BEWAG–Datenausstrahlung über DCF49.* persönliche Mitteilung, 1993.

[Eve93] H. Eveking. *Verifikation.* Teubner Studienbücher, 1993.

[fdF90] Vollzugsordnung für den Funkdienst. *Band 2.* Anhang 6, 1990.

[Fet86] A. Fettweis. *Wave Digital Filters: Theorie and Practice. Proceedings of the IEEE*, Seiten 270–327, Februar 1986.

[FR88] W. Fumy und H. Rieß. *Kryptographie.* R. Oldenbourg Verlag, 1988.

[Fra92] L. Franks. *Carrier and Bit Synchronization in Data Communication — A Tutorial Review. IEEE Transactions on Communications*, Seiten 1107–1120, August 1992.

[Gar79] F. Gardner. *Phaselock Techniques.* John Wiley, 1979.

[Gar88] F. Gardner. *Demodulator Reference Recovery Techniques Suited for Digital Implementation. European Space Agency*, Seiten 306–367, August 1988. Final Report, ESTEC.

[GC77] D. J. Goodman und M. J. Carey. *Nine Digital Filters for Decimation and Interpolation. IEEE Transactions on Acoustics, Speech and Signal Processing*, Seiten 121–126, April 1977.

[Ges84] H. Geschwinde. *Einführung in die PLL–Technik.* Vieweg, 1984.

[GHB93] J. Greene, E. Hamdy und S. Beal. *Antifuse Field Programmable Gate Arrays. Proceedings of the IEEE*, Seiten 1042–56, July 1993.

[GHMB93] Y. Gao, D. Herold und U. Meyer-Bäse. *Zum bestehenden bertragungsprotokoll kompatible Fehlerkorrektur.* In *Funkuhren Zeitsignale Normalfrequenzen*, Seiten 99–112, Mai 1993.

[Goo91] F. Goodenough. *ISSCC Analog Technology. Electronic Design*, Seiten 67–86, Februar 1991.

[Gro91] B. Gross. *Verbesserung der Ton/Datenübertragung bei der D2–MAC–Fernsehnorm.* Dissertationsschrift, Institut für Übertragungstechnik und Elektroakustik, THD, 1991.

[Hah91] H. Hahn. *Untersuchung und Integration von Berechnungsverfahren elementarer Funktionen auf CORDIC–Basis mit Anwendungen in der adaptiven Signalverarbeitung.* Dissertationsschrift, VDI Verlag, Reihe 9, Nr. 125, 1991.

[HAHL85] J. Haase, E. Auer, K. Höfer und E. Lüder. *A High Speed Parallel Architecture for Recursive Digital Filtering. Archiv der elektrischen Übertragung (AEÜ)*, Seiten 241–246, Band 40, 1985.

[Har90] Harris. *Datenblatt.* HSP45116 Numerically controlled Oscillator/Modulator, 1990.

[Har92] Harris. *Datenbuch.* Digital Signal Processing, 1992.

[Her69a] O. Herrmann. *Quadraturfilter mit rationalem Übertragungsfaktor. Archiv der elektrischen Übertragung (AEÜ)*, Seiten 77–84, Band 23, 1969.

[Her69b] O. Herrmann. *Transversalfilter zur Hilbert-Transformation. Archiv der elektrischen Übertragung (AEÜ)*, Seiten 581–587, Dezember 1969.

[Her91b] R. Hersam. *Durchbruch bei rauscharmer Frequenzsynthese. Elektronik Industrie*, April 1991.

[Hes89] W. Hess. *Digitale Filter.* Teubner Studienbücher, 1989.

[Het87] P. Hetzel. *Zeitübertragung auf Langwelle durch amplitudenmodulierte Zeitsignale und pseudozufällige Umtastung der Trägerphase.* Dissertationsschrift, Institut für Zeitmeßtechnik, Fein– und Mikrotechnik der Universität Stuttgart, 1987.

[Hil83] W. Hilberg. *Funkuhren.* R. Oldenbourg Verlag, 1983.

[Hil88] W. Hilberg. *Funkuhrtechnik.* R. Oldenbourg Verlag, 1988.

[Hil89] W. Hilberg. *Grundlagen elektronischer Schaltungen.* R. Oldenbourg Verlag, 1989.

[Hil92a] W. Hilberg. *Redundanz im DCF77 Format.* persönliche Mitteilung, 1992.

[Hil93] W. Hilberg. *Funkuhren Zeitsignale Normalfrequenzen.* Verlag Sprache und Technik, 1993.

[HM88] H. Hikawa und S. Mori. *A digital frequency synthesizer with phase accumulator. IEEE International Symposium on Circuits and Systems, Proceedings ESPOO*, Seiten 373–376, Juni 1988.

[HMB94] W. Hilberg und U. Meyer-Bäse. *Heimische Wetterstation.* Patentanmeldung, Februar 1994.

[Hog81] E. B. Hogenauer. *An Economical Class of Digital Filters for Decimation and Interpolation. IEEE Transactions on Acoustics, Speech and Signal Processing*, Seiten 155–162, April 1981.

[Hol82] J. Holmes. *Tracking –Loop Bias Due to Costas Loop Arm Filter Imbalance. IEEE Transactions on Communications*, Seiten 2271–3, Oktober 1982.

[Hu92] Yu Hen Hu. *The Quantization Effects of the CORDIC–Algorithm. IEEE Transactions on signal processing*, Seiten 834–844, April 1992.

[Joh75] R. Johannesson. *Robustly Optimal One–Half Binary Convolutional Codes. IEEE Transactions on Information Theory*, Seiten 464–8, Juli 1975.

[Kal80] R. Kalhöfer. *Die Realisierung kompakter störsicherer Funkuhren mit den Mitteln der Mikroelektronik.* Dissertationsschrift, Institut für Datentechnik, THD, 1980.

[Ker89] H. Kerler. *Cityruf. Funkschau*, Seiten 19–21, 1 / 1989.

[KK89] K.D. Kammeyer und K. Kroschel. *Digitale Signalverarbeitung.* Teubner Studienbücher, 1989.

[Kri74] L. Kristiansson. *Jump Phenomenon in Digital Filters. Electronics Letters*, Seiten 14–15, Januar 1974.

[Kü74] K. Küpfmüller. *Die Systemtheorie der elektrischen Nachrichtenübertragung.* S. Hirzel Verlag, 1974.

[LC81] W. Lindsey und C. Chie. *A Survey of Digital Phase–Locked Loops. Proceedings of the IEEE*, Seiten 410–431, April 1981.

[LH93] L. Longo und B.-R. Horng. *A 15b 30kHz Bandpass Sigma-Delta Modulator. IEEE International Solid-State Circuits Conference*, Seiten 226–7, Februar 1993.

[Lin74] H. Lindemeier. *Die transistorierte Empfangsantenne mit kapazitiv hochohmigem Verärker als optimale Lösung für den Empfang niedriger Frequenzen. NTZ Archiv*, Seiten 227–233, 1974.

[LP73] T. Lewis und W. Payne. *Generalized Feedback Shift Register Pseudorandom Number Algorithm. Journal of the Association for Computing Machinery*, Seiten 456–458, Juli 1973.

[LS87] W. L. Lee und C. G. Sodini. *A Topology for Higher Order Interpolative Coders.* In *Proceedings Internatinal Symposium Circuits and Systems*, Seiten 459–462, Mai 1987.

[LSS88] V.G. Luchkov, S.V. Shamshin und V.V. Shakhgilgyan. *Digital Calculating Frequency Synthesizer with Full Compensation of Phase Errors. Telecommun. Radio Eng.*, Seiten 71–74, August 1988.

[Lü78] H. D. Lüke. *Zur Entstehung des Abtasttheorems. NTZ*, Seiten 271–274, April 1978.

[Lü88] H. D. Lüke. *Signalübertragung.* Springer–Lehrbuch, 1988.

[M+91] M. Mayers et al. *Alles über AD-DA-Wandler. Design & Elektronik*, Seiten 30–50, Juli 1991.

[Mar82] H. Marko. *Methoden der Systemtheorie.* Springer Verlag, 1982.

[MB89] U. Meyer-Bäse. *Untersuchung einer schwellenverbesserten Begrenzer-Diskriminatordemodulation von digitalen FM–Signalen mit Hilfe von Rechnersimulationen.* Diplomarbeit DT 2477, Institut für Übertragungstechnik und Elektroakustik, THD, 1989.

[MB91] U. Meyer-Bäse. *Entwurf und Untersuchung universeller digitaler Empfängerprinzipien.* Institutsbericht 135, Institut für Datentechnik, THD, 1991.

[MB92a] U. Meyer-Bäse. *Ein für Bandpaßsignale optimierter ΣΔ–A/D–Wandler.* Institutsbericht 158, Institut für Datentechnik, THD, 1992.

[MB92b] U. Meyer-Bäse. *Neuer hybrider Frequenzsynthesizer mit Frequenzfehlerkompensation.* Institutsbericht 147, Institut für Datentechnik, THD, 1992.

[MB93a] U. Meyer-Bäse. *Eine neue Klasse von Frequenzabtastfilter reduzierter Komplexität.* Institutsbericht 181, Institut für Datentechnik, THD, 1993.

[MB93b] U. Meyer-Bäse. *Universeller Hilbert–Abtastempfänger mit CORDIC-Demodulation.* In *Funkuhren Zeitsignale Normalfrequenzen*, Seiten 65–81, Mai 1993.

[MB+94] U. Meyer-Bäse et al. *COordinate Rotation DIgital Computer (CORDIC) Synthesis for FPGA.* In *Field Programmable Logic and Applications, Prag*, Seiten 397–408, September 1994.

[MBH92] U. Meyer-Bäse und W. Hilberg. *Schmalbandempfänger für Digitalsignale.* Deutsches Patent DE 4219417.2, Juni 1992.

[MBH94a] U. Meyer-Bäse und W. Hilberg. *Bandpaß–ΣΔ–A/D–Wandler für Langwellendatenempfänger.* Patentanmeldung in Vorbereitung, 1994.

[MBH94b] U. Meyer-Bäse und W. Hilberg. *Paging–System für Langwelle.* Patentanmeldung, Mai 1994.

[MBW94a] U. Meyer-Bäse und S. Wolf. *Neuartiger Akkumulatorsynthesizer mit Frequenzfehlerkompensation.* In *Thesys Wettbewerbsbeitrag: Chip Idee des Jahres 1994*, Seiten 1–10, September 1994.

[MBW94b] U. Meyer-Bäse und S. Wolf. *Neuartiger FPGA basierter Akkumulatorsynthesizer mit Gigahertz Performance.* In *GI/GME/ITG–Fachtagung, Rechnergestützter Entwurf und Architektur mikroelektronischer Systeme*, Seiten 228–231, Mai 1994.

[MC71] J. Massey und D. Costello. *Nonsystematic Convolutional Codes for Sequential Decoding in Space Applications. IEEE Transactions on Communications*, Seiten 806–813, Oktober 1971.

[McC92] D. McCarty. *Digital PLL Suits FPGAs. Elektronic Design*, Seite 81, März 1992.

[MEA89] K. Manivannan, C. Eswaran und A. Antoniou. *Design of Low-Sensitivity Universal Wave Digital Biquads. IEEE Transactions on Circuits and Systems*, Seiten 108–113, Januar 1989.

[MG92] H. Meinke und F. Gundlach. *Taschenbuch der Hochfrequenztechnik.* Springer Verlag, 1992.

[ML78] G. O. Martens und H. H. Lê. *Wave Digital Adaptors for Reciprocal Second-Order Sections. IEEE Transactions on Circuits and Systems*, Seiten 1077–1083, Dezember 1978.

[MM80] P. R. Moon und G. O. Martens. *A Digital Filter Structure Requiring Only m-Bit Delays, Shifters, Inverters and m-Bit Adders Plus Simple Logic Circuitry. IEEE Transactions on Circuits and Systems*, Seiten 901–908, Oktober 1980.

[MMR78] W. Mills, C. Mullis und R. Roberts. *Digital Filter Realizations without Overflow Oscillations. IEEE Transactions on Acoustics, Speech and Signal Processing*, Seiten 334–338, August 1978.

[MS76] F. MacWilliams und J. Sloane. *Pseudo–Random Sequences and Arrays. Proceedings of the IEEE*, Seiten 1715–29, Dezember 1976.

[MT81] A. Makarios und T. Tozer. *False–Look Avoidance Scheme for Costas Loops. Electronics Letters*, Seiten 490–2, Juli 1981.

[MT82] A. Mararios und T. Tozer. *False–Lock Performance Improvement in Costas Loops. IEEE Transactions on Communications*, Seiten 2285–88, Oktober 1982.

[Nag90] A. Nagase. *Eine Analyse des ASIC–Marktes. Elektronik Industrie*, Seiten 150–158, November 1990.

[NHP81] S. Nishimura, K. Hirano und R. N. Pal. *A New Class of Very Low Sensitivity and Low Roundoff Noise Recursive Digital Filter Structures. IEEE Transactions on Circuits and Systems*, Seiten 1152–1158, Dezember 1981.

[Nie92] E. Nienhaus. *Maximale Datenraten für die Langwellensender.* persönliche Mitteilung, 1992.

[Nie93] B. Niedermeier. *Frequenzerzeugung mit schnellen EPLDs. Elektronik Industrie*, Seiten 42–44, Dezember 1993.

[NS91] H.T. Nicholas und H. Samuelli. *A 150-MHz Direct Frequency Synthesizer in 1.25µ CMOS with -90dBc Spurious Performance. ISSCC*, Seiten 42–43, Februar 1991.

[OL74] W. Osborne und M. Luntz. *Coherent and Noncoherent Detection of CPFSK. IEEE Transactions on Communications*, Seiten 1023–1036, August 1974.

[OS92] A. V. Oppenheim und R. W. Schafer. *Zeitdiskrete Signalverarbeitung.* Oldenbourg Verlag, 1992.

[Osb82] H. Osborne. *A Generalized "Polarity–Type" Costas Loop for Tracking MPSK Signals. IEEE Transactions on Communications*, Seiten 2289–96, Oktober 1982.

[P+91] R. Plassche et al. *38th ISSCC Session 4: Oversampling Converts.* In *Digest of Technical Papers*, Seiten 68–75, Februar 1991.

[P+92a] W. H. Press et al. *Numerical Recipes in C.* Cambride University Press, 1992.

[P+92b] S. Prutton et al. *Was Wandler wirktlich können. Design & Elektronik*, Seiten 42–67, August 1992.

[Pap86] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw–Hill, 1986.

[PF88] H. Padir und L. E. Franks. *An New Digital Recursive Filter Structure with Low Roundoff Noise and Low Limit Cycle Oscillations. Twenty-Second Asilomar Conference on Signals, Systems and Computers*, Seiten 137–141, Oktober 1988.

[Pic91] H. Pichlik. *Architekturwettbewerb. ELRAD*, Seiten 40–50, April 1991.

[Ple89] GEC Plessey. *Datenblatt.* SP2002 Direct Frequency Synthesizer, 1989.

[Ple90a] Plessey. *Elektrisch rekonfigurierbares Array. Elektronik Industrie*, Seiten 16–18, August 1990.

[Ple90b] GEC Plessey. *Datenbuch.* Digital Signal Processing IC Handbook, 1990.

[PO88] G.N. Prokhladin und M.Y. Osetrov. *Calculation of the Transient in Digital Frequency Synthesizers. Telecommun. Radio Eng.*, Seiten 153–156, Oktober 1988.

[Pro83] J. Proakis. *Digital Communications.* McGraw–Hill, 1983.

[Qua91] Qualcomm. *Datenblatt.* Q2334 Direct Digital Synthesizer, 1991.

[Rie93] P. Riekert. *Stromverbrauch im Vergleich. Design & Elektronik*, Seiten 94–95, Dezember 1993.

[Sch74] H. Schmid. *Decimal Computation.* John Wiley & Sons, 1974.

[Sch84] H.W. Schüssler. *Netzwerke, Signale und Systeme.* Springer Verlag, 1984.

[SH90] S. D. Stearns und D. R. Hush. *Digital Signal Analysis.* Prentice Hall, 1990.

[She90] T. Sheffer. *Introduction to direct digital synthesis. Proceedings RF Technology, EXPO*, Seiten 401–411, März 1990.

[Sin86] V. Singh. *On the Realization of Two's Complement Overflow Limit Cycle Free State-Space Digital Filters. Proceedings of the IEEE*, Seiten 1287–1288, September 1986.

[SJF92] M. Snelgrove S. Jantzi und P. Ferguson. *A 4th-Order Bandpass Sigma-Delta Modulator. Proceedings Custom Integrated Circuits Conference*, Seiten 16.5.1–4, Mai 1992.

[Skl88] B. Sklar. *Digital Communications.* Prentice Hall, 1988.

[SP83] P. K. Sim und K. K. Pang. *On the Asymtotic Stability of an Nth Order Nonlinear Recursive Digital Filter. IREECON 19th International Electronics Convention and Exhibition*, Seiten 108–110, September 1983.

[Spa76] O. Spaniol. *Arithmetik in Rechenanlagen.* Teubner Studienbücher, 1976.

[Spe94] G. Specht. *Forschungs- & Technologiemanagement.* Vorlesungs–Skript, WS 1993/94.

[SR64] R. Sanneman und J. Rowbotham. *Unlock Characteristics of the Optimum Type II Phase-Locked Loop. IEEE Transactions on Aerospace and navigational Electronics*, Seiten 15–24, März 1964.

[SS90] A. Sherstinsky und C. G. Sodini. *A Programmable Demodulator for Oversampled Analog-to-Digital Modulators. IEEE Transactions on Circuits and Systems*, Seiten 1092–1103, September 1990.

[Sta73] W. Stahnke. *Primitive Binary Polynomials. Mathematics of Computation*, Seiten 977–980, Oktober 1973.

[Ste88] J. Stensby. *False Lock in Costas Loops. Proceedings of the 20. Southeastern Symposium on System Theory*, Seiten 75–79, März 1988.

[SVGR93] A. Sangiovanni-Vincentelli, A. Gamal und J. Rose. *Synthesis Methods for Field Programmable Gate Arrays. Proceedings of the IEEE*, Seiten 1057–83, July 1993.

[SW90] D. Sautter und H. Weinerth. *Elektronik und Mikroelektronik.* VDI Verlag, 1990.

[SW92] K. Schmidt und A. Wittneben. *Systematic Complexity Reduction for Digital Square Timing Recovery. IEEE VTC*, Seiten 283–288, 1992.

[Swe91] P. Sweeney. *Error Control Coding.* Prentice Hall, 1991.

[T+92] Tröster et al. *An Interpolative Bandpass Converter on a 1.2μm BiCMOS Analog/Digital Array. Symposium on VLSI Circuits Digest of Technical Papers*, Seiten 102–3, Juni 1992.

[Tec92] Markt & Technik. *FPGAs und Segmented PLDs im steilen Flug nach oben.* Seiten 67-68, August 1992.

[Tel90] Stanford Telecom. *Datenblatt.* STEL-1177 Phase and Frequency modulated numerically controlled Oscillator, 1990.

[Tel91] Deutsche Bundespost Telekom. *Cityruf Funkempfänger — Betriebliche und technische Merkmale. FTZ-Richtlinie 171*, Juni 1991. TR 1, Zentralamt für Mobilfunk.

[TG] D. Troha und J. Gallia. *Digital phase-locked loop design using SN54/74LS297.* TI, Application Note AN3216.

[Tim90] D. Timmermann. *CORDIC-Algorithmen, Architekturen und monolithische Realisierungen mit Anwendungen in der Bildverarbeitung.* Dissertationsschrift, VDI Verlag, Reihe 10, Nr. 152, 1990.

[Tri93] S. Trimberger. *A Reprogrammable Gate Array and Applications. Proceedings of the IEEE*, Seiten 1030–41, July 1993.

[TS86] U. Tietze und Ch. Schenk. *Halbleiter–Schaltungstechnik.* Springer Verlag, 1986.

[UHKI88] K. Uchimura, T. Hayashi, T. Kimura und A. Iwata. *Oversampling A-to-D and D-to-A Converters with Multistage Noise Sharping Modulators. IEEE Transactions on Acoustics, Speech and Signal Processing*, Seiten 1899–1905, Dezember 1988.

[Unb86] H. Unbehauen. *Regelungstechnik I.* Vieweg, 1986.

[Ur81] H. Ur. *A New Methode for Limit Cycle Evaluation. IEEE International Symposium on Circuits and Systems*, Seiten 947–950, April 1981.

[Vit66] A. Viterbi. *Principles of Coherent Communication.* McGraw–Hill, 1966.

[Vol59] J. E. Volder. *The CORDIC Trigonometric computing technique. IRE Transactions on Electronics Computers*, Seiten 330–4, September 1959.

[Wal71] J.S. Walther. *A Unified algorithm for elementary functions. Spring Joint Computer Conference*, Seiten 379–385, Mai 1971.

[Wei85] A. Weinberg. *I/Q Reversal Phenomena in 4–Phase Modified Costas Loops. IEEE International Conference on Communications, Chicago*, Seiten 1015–9, Juni 1985.

[Wie88] J. Wietzke. *Kriterien zur einheitlichen Beurteilung der prinzipiellen Leistungsfähigkeit verschiedener Zeitzeichenempfänger unter Einbeziehung neuer digitaler Varianten.* Dissertationsschrift, Institut für Datentechnik, THD, 1988.

[Wie92a] J. Wietzke. *Division mit MDAC.* persönliche Mitteilung, 1992.

[Wie92b] J. Wietzke. *Eigenstörungen bei Null–ZF–Empfänger.* persönliche Mitteilung, 1992.

[Wig88] D. Wiggert. *Error–Control Coding and Applications.* Artech House, 1988.

[Xil93] Xilinx. *Datenbuch.* XC2000, XC3000 und XC4000, 1993.

[ZB87] O. Zinke und H. Brunswig. *Lehrbuch der Hochfrequenztechnik.* Springer–Verlag, 1987.

[ZL81] J. Zeman und A. G. Lindgren. *Fast Digital Filter with Low Round-Off Noise. IEEE Transactions on Circuits and Systems*, Seiten 716–723, Juli 1981.

Verzeichnis der betreuten Studien– und Diplomarbeiten

[Abd90] D. Abdolrahimi. *Entwurf und Aufbau eines Patterngenerators und Logicanalysators.* Studienarbeit st, Institut für Informationsverwaltung und Interaktive Systeme, 1990.

[Abd92] D. Abdolrahimi. *Implementierung von Software–Algorithmen für den Entwurf von digitalen Filtern mit beliebigem Amplitudengang.* Diplomarbeit DT 534, Institut für Datentechnik, THD, 1992.

[All92] E. Allmann. *Entwurf von digitalen Bandpaßfiltern für einen optimierten $\Sigma\Delta$–Bandpaß–A/D–Wandler.* Diplomarbeit DT 558, Institut für Datentechnik, THD, 1992.

[Ann92] B. Annamaier. *Implementierung von Software–Algorithmen für eine PC–Meßkarte.* Diplomarbeit DT 535, Institut für Datentechnik, THD, 1992.

[AW91] E. Allmann und T. Wolf. *Entwicklung und Aufbau eines vollständig digitalen Null–ZF–Empfängers unter Einsatz von programmierbaren Gate Arrays und Fließkomma DSPs.* Studienarbeit ST 493, Institut für Datentechnik, THD, 1991.

[BB92] R. Bach und C. Brandt. *Universeller Hilbert–Abtastempfänger mit CORDIC–Demodulation durch programmierbare Gate Arrays.* Studienarbeit ST 551, Institut für Datentechnik, THD, 1992.

[Bra90] M. Brauner. *Entwicklung eines universellen Wetter–FAX–Empfängers mit Rechnerinterface für die RS232–Schnittstelle.* Studienarbeit ST 480, Institut für Datentechnik, THD, 1990.

[Bug91] R. Bug. *Entwicklung und Aufbau eines frequenzselektiven Zweiwege–Fading–Simulators mit digitalen Signalprozessoren.* Diplomarbeit DT 517, Forschungsinstitut für Kommunikationstechnik der Robert Bosch GmbH, Hildesheim, 1991.

[Bur91] M. Burschel. *Aufbau und Inbetriebnahme eines Micro–Controller–Systems für einen universellen digitalen LW–Empfänger.* Studienarbeit ST 498, Institut für Datentechnik, THD, 1991.

[Die91a] H. Diehl. *Implementierung eines Table–Look–Up–Verfahrens für einen universellen Funkuhrempfänger.* Studienarbeit ST 499, Institut für Datentechnik, THD, 1991.

[Die91b] V. Dierkes. *Entwicklung und Aufbau einer Fehlerschutzeinheit für eine Satellitenübertragungsstrecke für digitale HDV–Signale.* Diplomarbeit DT 509, Forschungsinstitut für Kommunikationstechnik der Robert Bosch GmbH, Hildesheim, 1991.

[Die92] A. Dietrich. *Realisierung eines Hilbert-Abtastempfängers mit CORDIC-Demodulator für DCF77-Signale mittels eines Fließkomma-Signalprozessors.* Diplomarbeit DT 538, Institut für Datentechnik, THD, 1992.

[Fie90] W. Fieber. *Universeller Funkuhrempfänger mit indirekter Zeitauswertung durch ein Signalprozessorsystem.* Diplomarbeit DT 449, Institut für Datentechnik, THD, 1990.

[GS93] J. Guyot und H. Schmitt. *Entwicklung einer volldigitalen Costas-Loop mit programmierbaren Gate Arrays zu kohärenten Demodulation von Langwellendatensignalen.* Diplomarbeit DT 581, Institut für Datentechnik, THD, 1993.

[Hä93] T. Häuser. *Entwurf und Realisierung eines einheitlichen Bewertungsschemas für LW-Wetterbilder im TIFF-Format und Implementierung von Algorithmen zur Bildverbesserung auf einem PC-AT.* Diplomarbeit DT 619, Institut für Datentechnik, THD, 1993.

[Her91a] D. Herold. *Untersuchung von Maßnahmen zur Fehlerkorrektur für DCF77-Signale auf programmierbaren Gate Arrays.* Diplomarbeit DT 512, Institut für Datentechnik, THD, 1991.

[Heu93] T. Heute. *Verifikation von digitalen Frequenzabtastfiltern mittels einer parallelen FPGA/DSP-Struktur.* Diplomarbeit DT 600, Institut für Datentechnik, THD, 1993.

[HH90] D. Herold und R. Huthmann. *Funkdecoder für das Radio-Daten-System (RDS) mit dem Signalprozessor TMS320C25.* Studienarbeit ST 457, Institut für Datentechnik, THD, 1990.

[Hil92b] J. Hill. *Entwurf eines optimierten COordinate Digital Computer (CORDIC) mit programmierbaren Gate Arrays.* Diplomarbeit DT 550, Institut für Datentechnik, THD, 1992.

[HK91] T. Hattermann und M. Koch. *Optimierung von Lernstrategien für einen Tischtennis-Roboter.* Studienarbeit ST 479, Institut für Datentechnik, THD, 1991.

[HK92] H. Hausmann und V. Kleipa. *Aufbau eines Mikrocontroller-Systems und Entwicklung eines LCD-Graphik-Controllers auf FPGA-Basis.* Studienarbeit ST 520, Institut für Datentechnik, THD, 1992.

[HK93] T. Häuser und S. Kenne. *Entwicklung einer vollparallelen neuronalen CORDIC-Architektur mit Fließkomma-Signalprozessor und programmierbaren Gate Arrays.* Diplomarbeit DT 570, Institut für Datentechnik, THD, 1993.

[Hut92] R. Huthmann. *Entwurf eines für Bandpaß-Signale optimierten 1 Bit A/D-Wandlers.* Diplomarbeit DT 545, Institut für Datentechnik, THD, 1992.

[HW91] J. Hill und S. Wolf. *Entwicklung und Aufbau einer schnellen Signalprozessorkarte mit dem TMS320C30 DSP für AT-Computer.* Studienarbeit ST 494, Institut für Datentechnik, THD, 1991.

[Ker91] S. Kerkmann. *Untersuchung eines neuen Modulationsverfahrens mit Hilfe der hut-Funktion durch Rechnersimulation.* Studienarbeit ST 469, Institut für Datentechnik, THD, 1991.

[Kr"92] T. Krüger. *Entwicklung und Aufbau eines Mikrocontrollersystems zur Untersuchung neuartiger ferroelektrischer Speicherbausteine.* Diplomarbeit DT 559, Institut für Datentechnik, THD, 1992.

[Lei90] H. Leitel. *Rechnerschnittstelle für einen digitalen Video-Bildspeicher.* Studienarbeit st, Institut für elektrische Meßtechnik, 1990.

[Mai94] J. Maier. *Platinenkontrolle.* Diplomarbeit DT 000, Entwicklung Autoradio, Robert Bosch GmbH, Malaysia, 1994.

[MS93] J. Maier und T. Schubert. *Entwicklung eines Faltungsdecoders auf FPGA-Basis zur Fehlerkorrektur für ein Paging-System.* Studienarbeit ST 591, Institut für Datentechnik, THD, 1993.

[Nol89] A. Noll. *Entwicklung und Aufbau einer schnellen Bildverarbeitungseinheit zum Erkennen eines Objektes im Videobild.* Studienarbeit ST 416, Institut für Datentechnik, THD, 1989.

[Nol91] A. Noll. *Implementierung des CORDIC-Algorithmus zur Koordinatentransformation in einem programmierbaren Gate Array.* Diplomarbeit DT 481, Institut für Datentechnik, THD, 1991.

[Pra93] W. Praefcke. *Objektorientierte Analyse/Synthese-Codierung von Bewegtbildsequenzen.* Diplomarbeit DT 607, Forschungsinstitut für Kommunikationstechnik der Robert Bosch GmbH, Hildesheim, 1993.

[Res94] R. Resch. *Entwicklung und Aufbau von Modulen zur Emulation eines Autoradios und Erstellung einer graphischen Oberfläche zur Bedienung unter MS-Windows.* Diplomarbeit DT 626, Entwicklung Autoradio, Robert Bosch GmbH, Malaysia, 1994.

[RS93] R. Resch und P. Schreiner. *Entwicklung einer volldigitalen Phase-Locked-Loop mit programmierbaren Gate Arrays für Langwellen-FM-Enpfang.* Studienarbeit ST 595, Institut für Datentechnik, THD, 1993.

[Sch91a] C. Scheerer. *Entwicklung und Aufbau eines universellen digitalen LW-Empfängers.* Studienarbeit ST 467, Institut für Datentechnik, THD, 1991.

[Sch91b] D. Schulz. *Entwurf und Aufbau von 1-Bit-Digitalfiltern für Bandpaßsignale mit programmierbaren Gate Arrays.* Diplomarbeit DT 510, Institut für Datentechnik, THD, 1991.

[Sch91c] A. Schuppert. *Entwicklung aund Aufbau eines Bildspeichers für ein HDTV-CODEC.* Diplomarbeit DT 471, Forschungsinstitut für Kommunikationstechnik der Robert Bosch GmbH, Hildesheim, 1991.

[Sch92a] R. Schimpf. *Implementierung eines Neuronalen Netzes und Algorithmen zur Sprachvorverarbeitung auf einem Fließkommasignalprozessorsystem.* Diplomarbeit DT 553, Institut für Datentechnik, THD, 1992.

[Sch92b] B. Schlanske. *Entwicklung und Aufbau eines universellen Hilbert–Abtastempfängers mit CORDIC–Demodulation für LW–FAX–Signale durch digitalen Signalprozessor.* Diplomarbeit DT 537, Institut für Datentechnik, THD, 1992.

[Sch94a] H. Schmitt. *Entwicklung eines SCSI–Gerätetreibers zur Software–Anbindung eines 40MBit/s–Datenrecorders an eine SUN–Workstation.* Diplomarbeit DT 637, Institut für Datentechnik, THD, 1994.

[Sch94b] P. Schreiner. *Untersuchung neuartiger Verfahren des Diebstahlschutzes von Autoradios mit Hilfe von Transpondern.* Diplomarbeit DT 627, Entwicklung Autoradio, Robert Bosch GmbH, Malaysia, 1994.

[Sch94c] T. Schubert. *Konzipierung und Implementierung einer Novell–Netzwerkanwendung zur Prozeßverfolgung von Gerätevarianten in der Fertigung.* Diplomarbeit DT 000, Entwicklung Autoradio, Robert Bosch GmbH, Malaysia, 1994.

[Tam90] O. Tamm. *Wetter–FAX–Empfang mit digitaler Bildaufbereitung durch einen Personalcomputer.* Studienarbeit ST 456, Institut für Datentechnik, THD, 1990.

[Tam91] O. Tamm. *Entwicklung und Aufbau einer universellen Gate Array Platine und Implementierung eines Frequenzsynthesealgorithmus.* Diplomarbeit DT 476, Institut für Datentechnik, THD, 1991.

[Ull91] S. Ullrich. *Entwicklung und Aufbau einer Bandaufspaltung für HDTV–Signale mit Quadrature Mirror Filter.* Diplomarbeit DT 470, Forschungsinstitut für Kommunikationstechnik der Robert Bosch GmbH, Hildesheim, 1991.

[Wat93a] R. Watzel. *Entwurf eines Paging–Protokolls und Implementierung eines zugehörigen Krypto–Controllers auf FPGA–Basis.* Studienarbeit ST 590, Institut für Datentechnik, THD, 1993.

[Wat93b] R. Watzel. *Implementierung und Verbesserung von Lernalgorithmen der bekannten Radialbasisklassifikatoren und Anwendung eines modifizierten COordinate Rotation DIgital Computers (CORDIC).* Diplomarbeit DT 615, Institut für Datentechnik, THD, 1993.

[Wec93] H. Wech. *Entwicklung und Aufbau eines 40 MBit/s Datenrekorders mit variabler Datenbreite und intelligenter SCSI–Schnittstelle.* Diplomarbeit DT 594, Forschungsinstitut für Kommunikationstechnik der Robert Bosch GmbH, Hildesheim, 1993.

[Wol92a] S. Wolf. *Untersuchung eines neuen hybriden Frequenzsynthesizers mit Frequenzfehlerkompensation.* Diplomarbeit DT 529, Institut für Datentechnik, THD, 1992.

[Wol92b] T. Wolf. *Entwicklung eines modularen Konzepts zur Simulation von digitalen Übertragungsstrecken auf einer TMS320C30–Signalprozessorkarte.* Diplomarbeit DT 557, Institut für Datentechnik, THD, 1992.

[Zah90] F. Zahn. *Entwicklung 8 paralleler VLC–Datendekodierungseinheiten für einen HDTV–Codec zur Satellitenübertragung.* Diplomarbeit DT 455, Forschungsinstitut für Kommunikationstechnik der Robert Bosch GmbH, Hildesheim, 1990.

Abkürzungsverzeichnis

ADC	Analog to Digital Converter
ADPLL	vollständig digitale PLL
AGC	Automatische Verstärkungsregelung
ASIC	Anwender spezifischer Schaltkreis
ASK	Amplitude Shift Keying
AWGN	Additives weißes Gaußches Rauschen
BCD	Binary Coded Decimal
BCH	Bose–Chaudhuri–Hocquenghem (Code)
BDD	Binary Decision Diagramm
BFW	Bitfehlerwahrscheinlichkeit
BP	Bandpaß
BPSK	Binary PSK
BS	Barrel Shifter
CBPC	Cascaded BP Comb
CLB	Configurable Logic Block
CPLD	Komplexe Programmierbare Logikbausteine
CSD	Canonical Signed Digit, quasiternäres Zahlensystem
CSOC	kanonische selbstorthogonale Codes
CTPC	Cascaded TP Comb
DAC	Digital to Analog Converter
DCF	D=Deutschland; F=Nähe zu Frankfurt a. M.;C=Langwelle
DCO	digital steuerbarer Oszillatoren
DDS	Direkter Digitaler Synthesizer
DPLL	Digitale PLL mit analogen Zwischensignalen
DSP	Digitaler Signalprozessor
DWD	Deutscher Wetter-Dienst
EPROM	Erasable PROM
F1C	LW–FSK–Modulation Hub =±150 Hz
F3C	LW–MFSK–Modulation Hub =±150 Hz
FGI	Frequenzgrundimpuls
FIFO	First In — First Out (Modus)
FPGA	Field Programmable Gate Array
FRS	Forced Response Stability
FSF	Frequenzabtastfilter
FSK	Frequency Shift Keying
FSM	Finite State Machine
FU	Funkuhr
GRP	Global Routing Pool
HA0	Hilbert–Abtaster nullter Ordnung
HA1	Hilbert–Abtaster erster Ordnung
HBG	schweizer Funkuhrsender
IC	integrierter Schaltkreis
LCD	Liquid Crystal Devices (Anzeige)
LMS	(Verfahren der) kleinsten Quadrate

LRS	Links/Rechts–Shifter
LSB	Least Significant Bit
LUT	Look–Up Table
LW	Langwelle (30-300 kHz)
MC	Mikrocontroller
MDAC	Multiplizierender DAC
MFSK	Mehrstufige FSK
MLSE	Maximum Likelihood Sequence Estimation
MNF	Minimum Norm Filter
MPSK	Multiple PSK
MPX	Multiplexer
MSB	Most Significant Bit
NTF	Rauschübertragungsfunktion
PAM	Pulse Amplitude Modulation
PD	Phasendetektor
PDM	Pulsedauermodulation
PGA	Programmable Gate Array
PIA	Programmable Interconnect Array
PLIC	Programmierbare Logik IC
PLL	Phase–Locked Loop
PN	Pseudonoise (Sequence)
PREP	Programmable Electronics Performance Corporation
PROM	Programmable Read Only Memory
PSK	Phase Shift Keying
QAM	Quadraturamplitudenmodulation
QLI	Quick Look In (Codes)
RDS	Radio–Daten–System
RS	Rechts–Shifter
SPLD	einfache programmierbare Logikbausteine
SRAM	Static Random Access Memory
STF	Signalübertragungsfunktion
TFF	Toggle Flip–Flop
TP	Tiefpaß
TTL	Transistor–Transistor–Logik
UHF	Ultra High Frequencies (0,3-3 GHz)
UTC	Universal Time Coordinated
VCO	spannungsgesteuerter Oszillator
VGA	Video Graphics Adapter
VHDL	VHSIC Hardware Desription Language
VHF	Very High Frequencies (30-300 MHz)
VLF	Very Low Frequencies (3-30 kHz)
WD	Wellendigitalfilter
ZF	Zwischenfrequenz
ZIOLC	Zero Input Overflow Limit Cycle

Lebenslauf Uwe Meyer–Bäse Juli 1994

Angaben zur Person

GEBURTSDATUM:	10.7.1964
GEBURTSORT:	Kassel
STAATSANGEHÖRIGKEIT:	deutsch
BEKENNTNIS:	evangelisch
FAMILIENSTAND:	verheiratet seit 31.5.1991 mit Dipl.-Ing. Anke Meyer–Bäse
VATER :	Dr. Gerhard Meyer–Bäse, Ingenieur
MUTTER :	Helga Meyer–Bäse, Lehrerin
GESCHWISTER:	Marlies Meyer–Bäse (1960), Dr. Karen Schidt–Bäse (1961)
ADRESSE :	Murstr. 10, 64347 Griesheim

Schulausbildung

1971-1973	Grundschule Sandershausen
1973-1974	Grundschule Baunatal
1974-1983	Friedrichsgymnasium Kassel

Universität

1984	TH Darmstadt: Studium der Elektrotechnik
1987	7 Wochen Fachpraktikum BTS Darmstadt: Vorentwicklung HDTV
1989	6 Wochen Fachpraktikum Siemens München: Entwicklung Digitales Mobilfunktelefon
1987-88	Übungsgruppenleitung in Informatik und Mathematik
11/1989	Diplom der TH Darmstadt, Fachrichtung Nachrichtentechnik
SS93	Lehrauftrag: „Digitale Verarbeitung analoger Signale“
SS94	Lehrauftrag: „Digitale Empfänger für Funkuhren“

Berufliche Praxis

1983-1984	Grundwehrdienst Bundeswehr
1990	Fachübersetzer des Buches “DISCRETE TIME SIGNAL PROCESSING“ für den Oldenbourg Verlags, München
1993-1994	Freier Mitarbeiter der Firma TEMIC, Heilbronn
Seit 12/89	Wissenschaftlicher Mitarbeiter am Institut für Datentechnik — Arbeitsgebiete: Komplexe Algorithmen der DVAS, Digitale Empfänger, FPGA–Anwendungen, Hardware für Neuronale Netze, Signalprozessoren