



Technische Universität Darmstadt

Institut für Datentechnik
Fachgebiet Mikroelektronische Systeme
Prof. Dr. Dr. h.c. mult. Manfred Glesner

Diplomarbeit

**FPGA-Realisierung eines
Offset-Lokaloszillators basierend auf
PLL- und DDS-Technologien**

Bearbeiter : Martin Kumm
Betreuer : Dr.-Ing. Peter Zipf, Dr.-Ing. Harald Klingbeil
Beginn : 17. Januar 2007
Ende : 17. Juli 2007

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und nur mit den nach der Prüfungsordnung der Technischen Universität Darmstadt zulässigen Hilfsmitteln angefertigt habe. Die verwendete Literatur ist im Literaturverzeichnis angegeben. Wörtlich oder sinngemäß übernommene Inhalte habe ich als solche kenntlich gemacht.

Darmstadt, im Juli 2007

Martin Kumm

Zusammenfassung

Die vorliegende Arbeit wurde im Rahmen der Kooperation des Fachgebietes *Mikroelektronische Systeme* (MES) der Technischen Universität Darmstadt mit der Abteilung *Hochfrequenztechnik und Beschleunigungsstrukturen* der Gesellschaft für Schwerionenforschung (GSI) erstellt.

Die GSI betreibt Beschleunigeranlagen für Ionenstrahlen, für deren Hochfrequenzanlagen zum Teil aufwendige Regelungssysteme notwendig sind. Zur Lösung dieser Regelungsaufgaben wurde in der Vergangenheit ein universelles System entwickelt, das neben einer analogen Vorverarbeitung auf digitalen DSP- und FPGA-Technologien basiert. Auf Grundlage existierender Baugruppen sollte für dieses sogenannte DSP-System ein neuer, digitaler Offset-Lokaloszillator (Offset-LO) unter Verwendung eines FPGA entwickelt werden.

Ein Offset-LO arbeitet nach dem Prinzip einer *phase locked loop* (PLL). Die notwendigen Grundlagen für die in dieser Arbeit verwendeten digitalen PLL-Algorithmen einschließlich der DDS-Technologie werden ausführlich hergeleitet. Ausgehend vom Modell einer zeitkontinuierlichen PLL wird ein Entwurfsverfahren zur Bestimmung der Reglerkonstanten einer zeitdiskreten PLL vorgestellt. Für die resultierende zeitdiskrete Typ-II-PLL wird der Stabilitätsbereich für beliebige Ordnungen der Übertragungsfunktion angegeben. Darüber hinaus wird das Verhalten im Zeit- und Frequenzbereich dargestellt. Weiterhin werden Formeln zur stationären Genauigkeit dieser PLL-Architektur entwickelt.

Der Fangbereich stellt ein wichtiges Kriterium dar, das den nutzbaren Frequenzbereich einer PLL vorgibt. Um diesen abschätzen zu können, wurde eine analytische Lösung für die Ausrastfrequenz hergeleitet. Des Weiteren konnte ein Verfahren entwickelt werden, das den Fangbereich einer digitalen PLL erweitert.

Aufgrund der Komplexität sowie der limitierten Ressourcen des verwendeten Altera[®]-Cyclone[®]-FPGAs, wurden einige Komponenten wie z.B. digitale Filter auf Algorithmenebene so optimiert, dass eine Implementierung ohne Multiplizierer möglich wurde. Des Weiteren wurden Pipeline-Methoden verwendet, um den erforderlichen Datendurchsatz zu erreichen.

Es wurden Simulationswerkzeuge genutzt, die zum einen die frühe Simulation eines abstrakten Modells zur Auslegung des Systems und zum anderen die Einbindung

von VHDL-Komponenten zur bitgenauen Validierung der resultierenden Implementierung erlauben. Hierzu wurde eine Bibliothek für Matlab[®]/Simulink[®] entwickelt, die Komponenten zur Simulation von analogen und digitalen PLLs beinhaltet.

Mit dieser Arbeit konnte gezeigt werden, dass eine leistungsstarke Offset-LO-Implementierung zur Verarbeitung hochfrequenter Signale mit der Anwendung der digitalen Signalverarbeitung auf FPGAs möglich ist. Wie abschließende Messungen zeigten, ließ sich das Verhalten des realen Systems mit Hilfe der Simulation sehr präzise vorhersagen.

Inhaltsverzeichnis

Zusammenfassung	v
Notation	xv
1 Hintergründe	1
1.1 Aufbau der Arbeit	1
1.2 Die Beschleunigeranlagen der GSI	1
1.3 Prinzip eines Synchrotrons	3
1.4 Regelungssysteme der Synchrotron-Hochfrequenz	5
1.5 Das DSP-System	6
1.5.1 Aufbau	6
1.5.2 Funktion des Offset-LO	8
1.5.3 Anforderungen an den Offset-LO	9
2 Grundlagen der Phase-Locked Loop	11
2.1 Funktionsweise einer PLL	11
2.2 Grundbegriffe	13
2.2.1 PLL-Struktur	13
2.2.2 Arbeitsbereich einer PLL	13
2.3 Grundgleichungen der zeitkontinuierlichen PLL	14
2.3.1 Modellierung der PLL mit Hilfe von Phasensignalen	14
2.3.2 Allgemeine Übertragungsfunktionen	15
2.3.3 Ordnung und Typ einer PLL	15
2.3.4 Übertragungsfunktionen mit Regler	16
2.3.5 Stabilität	18
2.3.6 Bodediagramm	18
2.3.7 Stationäre Genauigkeit	19
2.3.8 Zeitverhalten	22
2.4 Grundgleichungen der zeitdiskreten PLL	22
2.4.1 Berechnung der z-Übertragungsfunktion als Näherung für kontinuierliche Systeme	22
2.4.2 Direkte digitale Frequenzsynthese	25
2.4.3 Übertragungsfunktion der zeitdiskreten PLL	27
2.4.4 Übertragungsfunktion mit Regler	28

2.4.5	Stationäre Genauigkeit	30
2.4.6	Zeitverhalten	31
2.4.7	Stabilität	32
2.4.8	Bodediagramm	34
2.5	Fang- und Ausrastbereich	34
2.6	Einsatz einer PLL als Offset-LO	37
2.7	Grundlagen der Phasendetektion	38
2.7.1	Phasendetektoren für rechteckförmige Signale	38
2.7.2	Phasendetektoren für sinusförmige Signale	42
2.7.3	Erzeugung eines analytischen Signals	44
2.7.4	Erweiterung des PLL-Einfangbereichs durch Phase Unwrapping	47
3	Realisierungskonzepte	51
3.1	Beschreibung der zur Verfügung stehenden Komponenten	51
3.1.1	Das FPGA Interface Board	51
3.1.2	Das DDS-FIB Adapter Board	53
3.1.3	Das ADC/DAC-FIB Adapter Board	54
3.2	Realisierungskonzept eines Offset-LOs mit FIB und DDS-FAB	55
3.3	Realisierungskonzept eines Offset-LOs mit FIB und ADC/DAC-FAB	56
4	Implementierung	59
4.1	Entwurfsablauf	59
4.2	Interface zu ADC/DAC-FAB	60
4.3	Zahlendarstellung	61
4.4	Aufbau der ADPLL	61
4.5	Implementierung der Interpolationsfilter	62
4.5.1	Anforderungen an die Interpolationsfilter	62
4.5.2	Frequency Sampling Filter	63
4.5.3	Entwurf der Interpolationsfilter	67
4.6	Implementierung des analytischen Filters	68
4.7	Implementierung der komplexen Multiplikation	72
4.8	Implementierung der PLL-Komponenten	73
4.8.1	Phasendetektor	73
4.8.2	Direct Digital Synthesizer	74
4.8.3	PI-Regler	75
4.8.4	Tiefpassfilter	76
4.8.5	Phase Unwrap	78
4.9	Ressourcenbedarf und Verzögerungszeiten der Komponenten	81
5	Simulation	83
5.1	PLL-Bibliothek für Matlab/Simulink	83
5.2	Verifikation des zeitkontinuierlichen Phasenmodells	86

5.3	Simulation der Offset-ADPLL mit implementierten VHDL Komponenten	87
6	Ergebnisse	95
6.1	Messergebnisse	95
6.1.1	Phasenauflösung des DSP-Systems	95
6.1.2	Messung der Sprungantwort des Offset-LO	96
6.1.3	Messung eines phasenmodulierten Signals	97
6.1.4	Phasenfehler bei frequenzdynamischen Eingangssignalen	98
6.2	Ausblick	99
	Literaturverzeichnis	101
7	Abkürzungen	105

Tabellenverzeichnis

1.1	Technische Daten SIS12/18 [Fra87]	4
1.2	Anforderungen an den Offset-LO	9
2.1	Kontinuierliche Eingangssignalformen [IB05]	21
2.2	SFDR-Werte einer DDS in Abhängigkeit von der Phasenauflösung [NS87]	27
2.3	Diskrete Eingangssignalformen [Unb00]	30
4.1	Koeffizienten der Polfilter mit Winkel der Polstelle bis zur Ordnung vier [MB04]	64
4.2	Bestimmung der Sinus-Quadranten	75
4.3	Auswertung der Phasendifferenz	80
4.4	Synthesergebnisse und Verzögerungszeiten der Komponenten	81
5.1	Realisierte Simulink-Modelle	84
5.2	Komponenten der Simulink [©] PLL-Bibliothek	85
5.3	Zusammenfassung der PLL Parameter	92

Abbildungsverzeichnis

1.1	Aufbau der bereits existierenden GSI-Gebäude und der zukünftigen FAIR-Anlage [G ⁺ 06]	2
1.2	Aufbau der analogen Vorverarbeitung des DSP-Systems	7
2.1	Aufbau einer PLL	12
2.2	Phasenmodell der analogen PLL	15
2.3	Bodediagramm der Führungsübertragungsfunktion	19
2.4	Bodediagramm der Fehlerübertragungsfunktion	20
2.5	Phasen- und Frequenz-Sprungantworten von $H(s)$ und $E(s)$	23
2.6	Aufbau eines DDS	25
2.7	Phasenmodell der zeitdiskreten PLL	28
2.8	Phasen- und Frequenz-Sprungantworten von $H(z)$ und $E(z)$	33
2.9	Stabilitätsdiagramm der zeitdiskreten Typ II PLL mit Ordnung D mit $\zeta=0,707$	33
2.10	Bodediagramm der Führungsübertragungsfunktion	34
2.11	Abhängigkeit der Ausrastfrequenz von ζ	36
2.12	Methoden zur Offset-Frequenzbildung	38
2.13	Aufbau des Phasen-Frequenz-Detektors	39
2.14	Aufbau des Binärzähler-Phasendetektors	40
2.15	Zeitverhalten des Binärzähler-Phasendetektors bei Phasenfehler	41
2.16	Zeitverhalten des Binärzähler-Phasendetektors bei Frequenzfehler	41
2.17	Eigenschaften eines FIR-Hilberttransformators 11. Ordnung	45
2.18	Auswirkung der Dämpfung des Hilbert-Filters auf den Phasenfehler	47
2.19	Aufbau des Phase-Unwrap	49
3.1	FPGA Interface Board	52
3.2	DDS FIB Adapter Board, aufgesteckt auf die FIB-Platine	53
3.3	ADC/DAC FIB Adapter Board	54
3.4	Aufbau mit Binärzähler-Phasendetektor	56
3.5	Aufbau der All-digital PLL	57
3.6	Aufbau der ADPLL-IQ Komponente	58
4.1	Gesamtaufbau der ADPLL	61
4.2	Aufbau der Entity <code>adpll.iq</code>	62

4.3	Aufbau der Abtastratenerhöhung	62
4.4	Spektren der um den Faktor 4 heraufgesetzten Eingangssignale	63
4.6	Abtastratenerhöhung mit FSF-Filtern	65
4.5	Betragsfrequenzgänge und Pol-/Nullstellen-Diagramm $F_{z6c3}(z)$	66
4.7	Spektren der zweistufigen Interpolation des HF-Signals	67
4.8	Spektren der zweistufigen Interpolation des ZF-Signals	69
4.9	Struktur der implementierten Abtastratenerhöhung	69
4.10	Betragsfrequenzgänge der HF- und ZF-Filter	70
4.11	Aufbau des Hilbert-Filters	71
4.12	Simulationsaufbau des analytischen Filters	72
4.13	Betragsfrequenzgang des implementierten analytischen Filters	73
4.14	Blockschaltbild der implementierten DDS	74
4.15	Struktur der Differenzgleichung des zeitdiskreten PI-Reglers	76
4.16	Blockschaltbild der Implementierung des diskreten PI-Reglers	76
4.17	Aufbau des diskreten PT1-Gliedes	77
4.18	Aufbau des diskreten multipliziererfreien PT1-Gliedes	78
4.19	Betragsfrequenzgang des resultierenden Tiefpassfilters	79
4.20	Aufbau der Phase-Unwrap Implementierung	79
5.1	Simulink-Blockschaltbild der analogen PLL mit idealem Phasendetektor	86
5.2	Phasenfehler bei einem Phasensprung von 180°	87
5.3	Phasenfehler und Ausgangsfrequenz bei verschiedenen Eingangsfre- quenzsprüngen	88
5.4	Simulink-Modell der Offset-ADPLL	89
5.5	FFT-Spektrum vor und nach Abtastratenerhöhung des HF-Signals . .	91
5.6	FFT-Spektrum vor und nach Abtastratenerhöhung des ZF-Signals . .	91
5.7	Simulationsergebnisse der Sprungantworten	93
6.1	Phasenauflösung des DSP-Systems	96
6.2	Messaufbau zur Bestimmung der Sprungantwort	97
6.3	Messung und Simulation eines Phasensprungs um 180°	97
6.4	Messung eines phasenmodulierten Signals	98
6.5	Messung des Phasenfehlers bei frequenzdynamischen Eingangssignalen	99

Notation

$a(t)$	Hüllkurve
A_B	gebrochene Binärzahl im Zweierkomplement-Format
A_D	Dezimaldarstellung von A_B
a_n	n-te Binärstelle von A_B
B	magnetische Flußdichte/magnetische Induktion
\dot{B}_{max}	maximale Induktionsänderung pro Zeiteinheit
B_{max}	maximale Induktion der Dipolmagnete
c_0	Lichtgeschwindigkeit (ca. $2,99792 \cdot 10^8$ m/s)
C	Umfang (engl. <i>circumference</i>)
D_{FB}	Verzögerung im Rückkopplungs-Pfad (engl. <i>feedback delay</i>)
D_{FF}	Verzögerung im Vorwärts-Pfad (engl. <i>feedforward delay</i>)
$E(s), E(z)$	Fehlerübertragungsfunktion
$\epsilon(t), \epsilon(n)$	Sprungfunktion
$e(t)$	Impulsantwort der Fehlerübertragungsfunktion
$F(s)$	Übertragungsfunktion eines Reglers
f_0	Umlauffrequenz
f_c	Mittenfrequenz des VCO (engl. <i>center frequency</i>)
$FTW(n)$	Frequenzwort einer DDS (engl. <i>frequency tuning word</i>)
$F_P(s)$	Übertragungsfunktion Proportional-Regler
$F_{PI}(s), F_{PI}(z)$	Übertragungsfunktion Proportional/Integral-Regler
$F_{RC}(s)$	Übertragungsfunktion RC-Filter
$F_{LL}(s)$	Übertragungsfunktion Lead-Lag-Filter

Notation

$F_{LP}(z)$	Übertragungsfunktion eines Tiefpassfilters
f_n	Eigenfrequenz
f_s	Abtastfrequenz (engl. <i>sampling frequency</i>)
f_{IF}	Zwischenfrequenz (engl. <i>intermediate frequency</i>)
f_{RF}	Hochfrequenz (engl. <i>radio frequency</i>)
f_o	Ausgangsfrequenz der PLL
f_g	Grenzfrequenz eines Filters
$\dot{f}_{RF,max}$	maximale Frequenzänderung der Hochfrequenz pro Zeiteinheit
Δf_{DDS}	Frequenzauflösung
$\varphi(t), \varphi(n)$	Phasensignal allgemein
$\varphi_i(t), \varphi_i(n)$	Eingangs-Phasensignal
$\varphi_o(t), \varphi_o(n)$	Ausgangs-Phasensignal
φ_R	Phasenreserve
$\varphi_{max,det}$	maximal detektierbare Phase des Phasendetektors
ϕ_0	Konstante Anfangsphase
ϕ_i	Konstanter Eingangsphasenoffset
ϕ_o	Konstanter Ausgangsphasenoffset
ϕ_s	Sollphase eines Teilchens
ϕ_{DDS}	Anfangsphase DDS
$\Phi_i(s), \Phi_i(z)$	Eingangsphase Bildbereich
$\Phi_o(s), \Phi_o(z)$	Ausgangsphase Bildbereich
$\Phi_{DDS}(z)$	Anfangsphase DDS Bildbereich
$\Delta\phi$	Phasendifferenz
$\Delta\Phi(s), \Delta\Phi(z)$	Differenzphase Bildbereich
$\Delta\varphi(t), \Delta\varphi(n)$	Differenz-Phasensignal
$\Delta\varphi_\infty$	Stationärer Phasenfehler
$G(s), G(z)$	Übertragungsfunktion allgemein
$g(t), g(n)$	Impulsantwort einer Übertragungsfunktion

h	Harmonischenzahl
$H(s), H(z)$	Führungsübertragungsfunktion
k_0	DDS-Verstärkung (zeitdiskret)
K_0	VCO-Verstärkung (zeitkontinuierlich)
k_d	Phasendetektor-Verstärkung (zeitdiskret)
K_d	Phasendetektor-Verstärkung (zeitkontinuierlich)
K_I	Verstärkung des integralen Regleranteils
K_P	Verstärkung des proportionalen Regleranteils
m	Masse
n	Index eines abgetasteten Signals bei $t = nT_s$
N_i	Eingangs-Frequenzteiler
N_o	Ausgangs-Frequenzteiler
O	Ordnung eines Filters
$PTW(n)$	Phasenwort einer DDS (engl. <i>phase tuning word</i>)
q	Ladung
r_M	Biegeradius der Dipolmagnete
r_R	mittlerer Radius eines Synchrotron-Rings
T_s	Periodendauer der Abtastung
T_L	Fangzeit einer PLL
$u_i(t), u_i(n)$	Eingangssignal
$u_o(t), u_o(n)$	Ausgangssignal
\hat{u}_i	Spitzenspannung des Eingangssignals
\hat{u}_o	Spitzenspannung des Ausgangssignals
$u_{RF}^{\hat{}}$	Spitzenspannung der Beschleunigungsspannung
u_{VCO}	VCO Regelsignal
v	Geschwindigkeit
ω_c	Mitten-Kreisfrequenz des VCO (engl. <i>center-frequency</i>)
ω_{DDS}	Mitten-Kreisfrequenz des DDS der zeitdiskreten PLL

Notation

ω_i	Eingangs-Kreisfrequenz
ω_m	Phasenmodulations-Kreisfrequenz
ω_n	Eigen-Kreisfrequenz
ω_o	Ausgangs-Kreisfrequenz
ω_r	Resonanz-Kreisfrequenz
ω_{RF}	Kreisfrequenz der Beschleunigungsspannung
$\Delta\omega$	Differenz-Kreisfrequenz
$\dot{\Delta\omega}$	Ableitung der Differenz-Kreisfrequenz $\Delta\omega$
$\Delta\omega_H$	Haltebereich (engl. <i>hold-in range</i>)
$\Delta\omega_L$	Fangbereich (engl. <i>lock-in range</i>)
$\Delta\omega_{PI}$	Ziehbereich (engl. <i>pull-in range</i>)
$\Delta\omega_{PO}$	Ausrastfrequenz (engl. <i>pull-out frequency</i>)
$\bar{x}(n)$	Mittelwert eines PWM-Signals
$\tilde{x}(t)$	geschätzter Mittelwert eines PWM-Signals
ζ	Dämpfung

1 Hintergründe

Ziel dieser Arbeit ist die FPGA-Realisierung eines Offset-Lokaloszillators (Offset-LO) basierend auf PLL- und DDS-Technologien. Dieser Offset-LO soll für Hochfrequenz-Regelsysteme der Beschleunigeranlagen der GSI eingesetzt werden.

1.1 Aufbau der Arbeit

Um die Hintergründe und Motivation dieser Arbeit zu verdeutlichen, sollen in den folgenden Unterkapiteln zunächst die Beschleunigeranlagen der GSI sowie die Funktionsweise eines Synchrotrons und die daraus resultierenden Anforderungen an die Hochfrequenzsysteme (HF-Systeme) vorgestellt werden. Anschließend werden die dafür benötigten HF-Regelsysteme sowie die Funktionsweise eines universellen DSP-Systems zur Lösung dieser Regelungsaufgaben erläutert. In diesem DSP-System arbeitet zurzeit ein analoger Offset-LO, der im Rahmen dieser Arbeit durch ein digitales Äquivalent ersetzt werden soll und über die Vorteile digitaler Schaltungen (Flexibilität, Reproduzierbarkeit, Abgleichfreiheit, etc.) verfügt. Es wird außerdem gezeigt, dass die Implementierung des Offset-LO als *Phase Locked Loop* (PLL) vorteilhaft ist. Daher ist das darauf folgende Kapitel den Grundlagen der PLL gewidmet, in dem die für die Ausarbeitung benötigten Grundlagen hergeleitet werden. Im anschließenden Kapitel werden mögliche Realisierungskonzepte mit existierenden Komponenten beschrieben. Es folgt je ein Kapitel über die Implementierung und Simulation des Offset-LO. Abschließend werden die Ergebnisse anhand von Messungen im letzten Kapitel dargestellt und zusammengefasst.

1.2 Die Beschleunigeranlagen der GSI

Die Gesellschaft für Schwerionenforschung (GSI) mit Sitz in Darmstadt ist Mitglied der Helmholtz-Gemeinschaft Deutscher Forschungszentren und Betreiber einer Beschleunigeranlage für schwere Ionen. Die Hauptforschungsgebiete der GSI sind Kern- und Atomphysik, Plasma- und Materialforschung sowie die Tumorthherapie im medizinischen Bereich. Die populärsten Forschungsergebnisse sind die Synthese von sechs

neuen chemischen Elementen mit den Ordnungszahlen 107 bis 112, sowie die Entwicklung einer neuartigen Tumorthherapie mit Ionenstrahlen. Aus den zur Gründung im Jahre 1969 geplanten 330 Mitarbeitern sind mittlerweile 1050 Beschäftigte geworden [GSI07]. Die Beschleunigeranlage besteht zurzeit aus dem Linearbeschleuniger UNILAC, dem Schwerionensynchrotron (SIS12/18¹) und dem Experimentier-Speicherring (ESR). Zurzeit arbeitet die GSI an der größten Erweiterung ihrer Beschleunigeranlagen, dem FAIR-Projekt. Bis 2015 sollen weitere Beschleuniger gebaut werden, deren größter Doppelring-Beschleuniger (SIS100/300²) einen Umfang von etwa 1100 m haben soll. Abbildung 1.1 gibt einen Überblick über die bestehende Anlage (linke Seite/blau) sowie den Endausbau der FAIR-Anlagen (rechte Seite/rot).

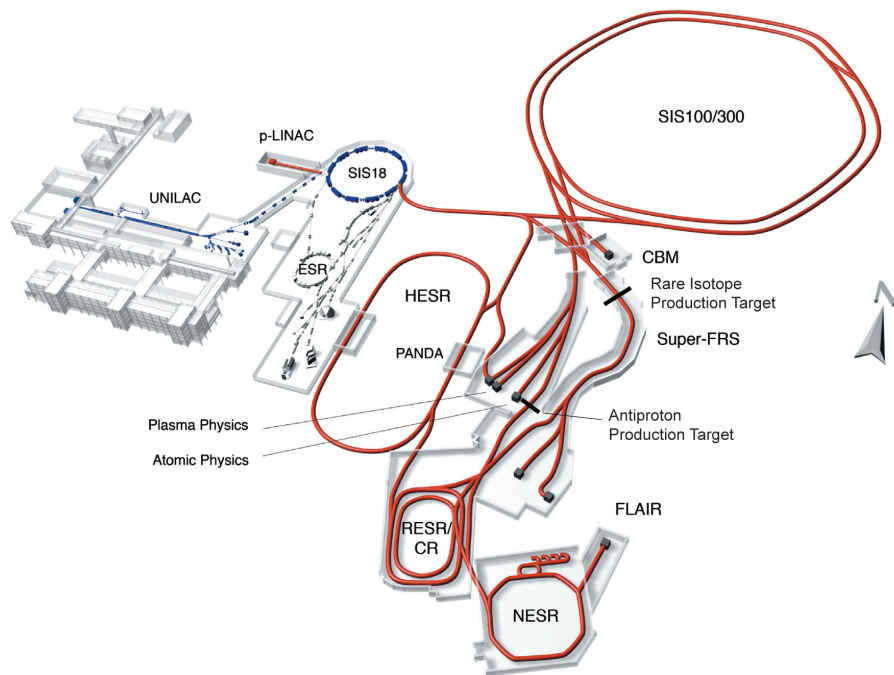


Abbildung 1.1: Aufbau der bereits existierenden GSI-Gebäude und der zukünftigen FAIR-Anlage [G⁺06]

Der Linearbeschleuniger UNILAC dient dem Synchrotron SIS12/18 als Vorbeschleuniger. Dieser beschleunigt die Ionen, die in den verfügbaren Quellen erzeugt werden, auf Energien von bis zu 20 MeV/u, abhängig von der Teilchensorte. Hinter dem UNILAC können bereits erste Experimente bei niedrigen Energien durchgeführt werden, oder der Teilchenstrahl wird zur weiteren Beschleunigung in das SIS12/18 injiziert. In Zukunft wird die Anlage um den Protonen-Linearbeschleuniger

¹Die Zahlen 12 und 18 stehen für die magnetische Steifigkeit von 12 Tm und 18 Tm

²Entsprechend der magnetischen Steifigkeit von 100 Tm bzw. 300 Tm.

(p-LINAC) ergänzt, was die Anlage um diese Teilchensorte erweitert. Nach Injektion in das SIS12/18 können die Teilchen auf ein Energieniveau von bis zu 2 GeV/u beschleunigt werden. Danach kann der Teilchenstrahl entweder direkt auf ein Experiment oder zur Speicherung in den ESR geleitet werden. Das SIS12/18 wird in Zukunft als Vorbeschleuniger für das SIS100/300 verwendet. In diesem sollen Energien von bis zu 34 GeV/u (U^{28+}) erreicht werden. Zusätzlich sollen weitere Speicherringe (HESR, RESR/CR und NESR) für verschiedene Experimente gebaut werden.

1.3 Prinzip eines Synchrotrons

Mit Ausnahme der Vorbeschleuniger (UNILAC und p-LINAC) gehören alle weiteren Beschleuniger zur Klasse der Synchrotrons mit starker Fokussierung. Diese bestehen aus drei wesentlichen Komponenten:

1. Dipolmagnete sorgen für eine Ablenkung des Teilchenstrahls auf einer ringförmigen Bahn.
2. Hochfrequenzkavitäten führen dem Teilchenstrahl über ein elektrisches Hochfrequenz-Feld die Energie zur Beschleunigung zu.
3. Zusätzliche fokussierende Magnete, sogenannte Quadropolmagnete, begrenzen die transversale Aufweitung des Teilchenstrahls.

Der Name Synchrotron leitet sich daraus ab, dass die Hochfrequenz der Kavitäten synchron zur Umlauffrequenz der Teilchen verändert wird. Dabei kann die Hochfrequenz ein Vielfaches (Harmonischnenzahl h) der Umlauffrequenz der Teilchen betragen:

$$f_{RF} = hf_0 \quad (1.1)$$

Das SIS12/18 wird momentan mit der Harmonischnenzahl $h = 4$ betrieben, als Vorbeschleuniger für das SIS100/300 soll $h = 2$ verwendet werden. Die Geschwindigkeit der Teilchen steht in linearem Zusammenhang zur Umlauffrequenz

$$v = 2\pi r_R f_0 \quad , \quad (1.2)$$

wobei r_R den mittleren Radius des Rings angibt. Der Energiegewinn der Teilchen ergibt sich aus der Beschleunigungsspannung zu

$$\Delta E = q\hat{u}_{RF} \sin(\omega_{RF}t + \phi_s) \quad , \quad (1.3)$$

wobei mit ϕ_s die Sollphase der Teilchen bezeichnet wird. Durch diesen Energiegewinn ergibt sich eine höhere Geschwindigkeit der Teilchen die gemäß (1.1) und (1.2) eine Anpassung der HF-Frequenz (ω_{RF}) erfordert. Die Dipolmagnete müssen dabei so

in ihrer magnetischen Induktion eingestellt werden, dass der mittlere Bahnradius der Teilchen konstant gehalten wird. Der Zusammenhang zwischen der magnetischen Induktion und der Teilchengeschwindigkeit ist durch

$$B = \frac{mv}{r_M q} \quad (1.4)$$

mit

$$m = \frac{m_0}{\sqrt{1 - (v/c_0)^2}} \quad (1.5)$$

gegeben, wobei r_M den magnetischen Biegeradius der Dipolmagnete, m_0 die Ruhemasse der Teilchen, m die (energieabhängige) Masse und c_0 die Lichtgeschwindigkeit bezeichnen.

Die Dipolmagnete sind in der Regel das begrenzende Element eines Synchrotrons, da die maximale magnetische Induktion B_{max} und die maximale Induktionsänderung pro Zeiteinheit \dot{B}_{max} begrenzt sind. Daher werden in der Regel alle anderen technischen Parameter von diesen abgeleitet. Der Zusammenhang zwischen der Frequenz der Beschleunigungsspannung und der magnetischen Induktion ergibt sich nach Umstellung und Zusammenfassung der letzten Gleichungen zu

$$f_{RF} = \frac{h}{2\pi r_R} \frac{B}{\sqrt{\left(\frac{m_0}{r_M q}\right)^2 + \left(\frac{B}{c_0}\right)^2}} \quad (1.6)$$

Nach dem Einsetzen der SIS12/18 Parameter (vgl. Tabelle 1.1) unter Berücksichtigung der unterschiedlichen Teilchensorten und Injektionsenergien ergibt sich ein maximaler Frequenzbereich von $f_{RF} = 0,8$ bis $5,4$ MHz. Für das SIS100/300 resultieren, aufgrund des größeren Ringumfangs, geringere Anforderungen an die Frequenzdynamik der Regelungssysteme ($h=10$, $f_{RF} = 1,1$ bis $2,7$ MHz).

Parameter	Formelzeichen	Wert
Maschinenumfang	C	216,72 m
Maximale Induktion der Dipolmagnete	B_{max}	1,8 T
Maximale Induktionsänderung	\dot{B}	10 T/s bis zu 1,2 T 4 T/s bis zu 1,8 T
Biegeradius der Dipolmagnete	r_M	10 m
Mittlerer Radius des Rings	r_R	34,5 m
Harmonischenzahl	h	4 (bzw. 2)

Tabelle 1.1: Technische Daten SIS12/18 [Fra87]

Ein weiterer für diese Arbeit wichtiger Parameter ist die maximale Frequenzänderung pro Zeiteinheit. Bei einer linear ansteigenden Induktion von $B(t) = \dot{B}t + B_0$ ergibt sich, mit Differenzieren von (1.6) nach der Zeit, die maximale Frequenzänderung zu

$$\dot{f}_{RF} = \frac{h\dot{B}m_0^2c_0^3}{2\pi r_{RF}r_m^2q^2} \left(\dot{B}^2t^2 + 2\dot{B}B_0t + B_0^2 + \left(\frac{m_0c_0}{r_mq} \right)^2 \right)^{-\frac{3}{2}}. \quad (1.7)$$

Diese ist maximal bei $t = 0$ und beträgt für SIS12/18 maximal $\dot{f}_{RF,max}=68,08$ MHz/s (für Protonen, $m_0 = 1,6726210^{-27}$ kg, $q = 1,6021810^{-19}$ C).

1.4 Regelungssysteme der Synchrotron-Hochfrequenz

Die Steuerung des gesamten Beschleunigers erfolgt vom zentralen Kontrollsystem. Dies verteilt alle notwendigen Steuerwerte wie z.B. die magnetische Induktion der Magnete, zeitlich aufgelöst an die einzelnen Geräte. Diese müssen wiederum dafür sorgen, dass diese Werte eingehalten werden. Im Hochfrequenzsystem des SIS12/18 existieren zwei Beschleunigungskavitäten, die jeweils eine Spitzenspannung von 16 kV erzeugen können. Da sowohl die Kavitäten selbst als auch die Verstärkerkette Frequenz- und Temperaturabhängigkeiten aufweisen, müssen sämtliche Sollwertvorgaben geregelt werden.

An dieser Stelle soll ein kurzer Überblick über die einzelnen HF-Regelungssysteme der SIS12/18 und ESR Anlage gegeben werden:

- **Amplitudenregelung:** Sie sorgt dafür, dass die Spitzenspannung der Kavität der Sollvorgabe entspricht. Die Beschleunigungsspannung der Kavität wird mit Hilfe eines Spannungsteilers reduziert, anschließend gleichgerichtet und mit dem Sollwert verglichen. Aus diesem Fehlersignal korrigiert der Regler das ansteuernde HF-Signal über einen Amplitudenmodulator.
- **Eigenfrequenzregelung:** Da die Kavitäten relativ schmalbandig bezogen auf ihren Frequenzbereich sind, müssen sie während der Beschleunigung fortlaufend in ihrer Resonanzfrequenz abgestimmt werden. Hierzu werden die Ferritringe der Kavität mit einem Gleichstrom vormagnetisiert. Diese Vormagnetisierung wird aus der Frequenzvorgabe abgeleitet und steuernd über eine Kennlinie vorgegeben. Der Eigenfrequenzregler misst dabei die Phasendifferenz zwischen Hochfrequenz-Strom und -Spannung und regelt diese über eine Korrektur des Vormagnetisierungsstroms zu Null, was der Resonanz der Kavität entspricht.

- **Kavitätensynchronisation:** Die relative Phasenlage der Hochfrequenzspannung der beiden Kavitäten muss entsprechend ihrer relativen Positionen im Ring eingestellt werden, damit sie beschleunigend wirkt. Dies wird durch Messung der Phasendifferenz und einer Frequenzkorrektur einer Kavität durch das Kavitätensynchronisationssystem erreicht.
- **Strahlphasenregelung:** Dieses Regelungssystem dient zur Dämpfung longitudinaler Schwingungen der Teilchenbündel. Die Phasenlage des Teilchenstrahls lässt sich aus dem Signal eines Strahlphasenmonitors detektieren und nach dem Herausfiltern der Synchrotronfrequenz als Regelgröße verwenden.

Es sei angemerkt, dass für die Beschleuniger des FAIR Projektes weit mehr Kavitäten zum Einsatz kommen³, die teilweise auf unterschiedlichen Harmonischen betrieben werden, was höhere Ansprüche an die Regelsysteme stellt. Außerdem sollen weitere Regelungssysteme zur Dämpfung von Teilchenschwingungen höherer Ordnung realisiert werden.

1.5 Das DSP-System

1.5.1 Aufbau

Wie im letzten Abschnitt gezeigt wurde, liegt bei fast allen regelungstechnischen Systemen der Synchrotron-Hochfrequenztechnik die Messung von Amplitude und Phase eines frequenzdynamischen Signals zugrunde. Um dieses Problem möglichst allgemein für alle Systeme zu lösen, wurde ein System, bestehend aus DSPs und FPGAs sowie einer analogen Vorverarbeitung entwickelt [Kli05]. Das Gesamtsystem wird im Folgenden als „DSP-System“ bezeichnet. Da die Sollfrequenz des Signals zu jedem Zeitpunkt genau bekannt ist, kann eine Frequenzverschiebung auf eine feste Zwischenfrequenz nach dem Heterodyne-Prinzip erfolgen.

Das Blockschaltbild des DSP-Systems ist in Abbildung 1.2 dargestellt. Es gibt zwei Messkanäle, „CH A“ und „CH B“, die beide gleichberechtigt sind. Einem weiteren Eingang „REF“ wird ein Referenz-Signal zugeführt, dessen Frequenz mit der Messfrequenz identisch ist. Dieses Signal wird aus dem gleichen HF-Generator gewonnen, der die Frequenz für die Beschleunigung vorgibt. Aus diesen Signalen werden zwei Zwischenfrequenz-Signale (ZF) erzeugt, die an den Analog/Digital-Umsetzern (ADC) abgetastet und anschließend vom DSP weiterverarbeitet werden. Das Wesentliche an diesem Konzept ist, dass beide frequenzkonstanten ZF-Signale jeweils die Phasen-

³Für SIS 100 werden z.B. 20 Kavitäten für die Beschleunigung verwendet.

und Amplitudeninformation der HF-Signale tragen. Im Folgenden wird das Blockschaltbild im Detail erklärt.

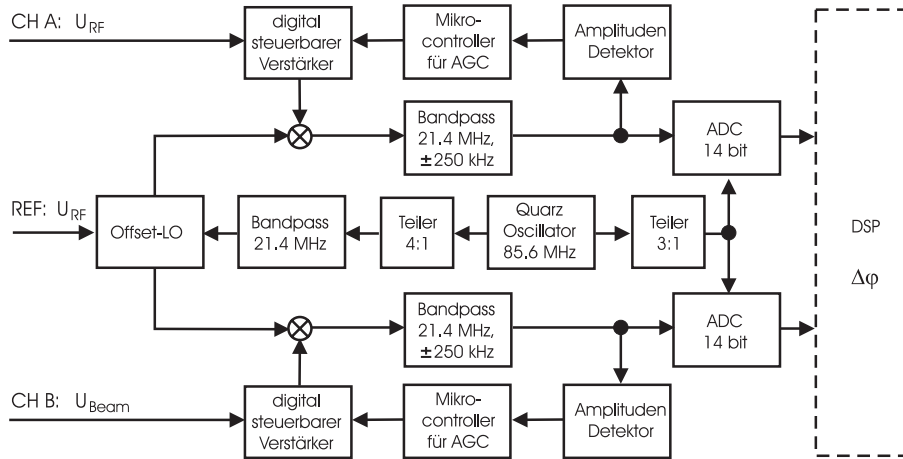


Abbildung 1.2: Aufbau der analogen Vorverarbeitung des DSP-Systems

Das Referenz-Signal wird zusammen mit dem konstanten Zwischenfrequenzsignal von $f_{IF} = 21,4$ MHz (engl. *intermediate frequency*, IF) auf einen Offset-Lokaloszillator (Offset-LO) gegeben, dessen Ausgangsfrequenz, der Summe der Frequenzen der beiden Eingangssignale entspricht ($f_{LO} = f_{RF} + f_{IF} = 22,2...26,8$ MHz). Dieses LO-Signal wird mit dem verstärkten Eingangssignal gemischt. Am Ausgang des Mixers entstehen neben parasitären Spektrallinien primär zwei Frequenzkomponenten. Ein Signal bei $f_{IF} + 2f_{RF}$ und ein Signal bei f_{IF} , welches durch ein Zwischenfrequenzfilter mit ± 250 kHz Bandbreite herausgefiltert wird. Die Amplitude des ZF-Signals wird anschließend detektiert und einer automatischen Verstärkungsregelung (engl. *Automatic Gain Control*, AGC) zugeführt, welche über den digital steuerbaren Verstärker für eine konstante Aussteuerung des ADCs sorgt.

Das ZF-Signal wird vom ADC unterabgetastet, d.h. die Zwischenfrequenz liegt oberhalb der Nyquist-Frequenz ($f_{IF} > f_s/2$). Dies ist zulässig, da die Bandbreite des ZF-Signals schmalbandig ist. Die Abtastfrequenz wird aus einem Quarzoszillator von 85,6 MHz mittels digitaler Teilung durch 3 abgeleitet. Aus dem gleichen Oszillator wird auch die Zwischenfrequenz mit einem Teiler von 4 abgeleitet. Die Abtastrate wird nochmals um den Faktor 19 im FPGA des ADC-Teils des DSP-Systems heruntersgesetzt ($f_s = 1,5017$ MHz). Damit entspricht das Verhältnis zwischen der Zwischen- und Abtastfrequenz genau $f_{IF}/f_s = (3 \cdot 19)/4 = 14,25$. Der gebrochene Anteil von $1/4$ führt dazu, dass genau 4 Abtastwerte einer ZF-Periode zugeordnet werden können und diese somit genau 90° auseinander liegen. Somit können diese direkt einem analytischen Signal (siehe Abschnitt 2.7.2 ab Seite 42) zugeordnet werden, was die Amplituden- und Phasendetektion enorm vereinfacht.

Die Algorithmen zur Detektion wurden zunächst auf dem DSP implementiert [Kli05]. Nach der erfolgreichen Erprobung in der Kavitätsynchronisation erfolgte eine Auslagerung dieser Algorithmen in das FPGA des DSP-Systems [GZMGK06], um die Funktionalität des DSP für die Strahlphasenregelung zu erweitern. Die DSP/FPGA-Platine hat den Formfaktor einer PCI-Karte und kann zur Diagnose von einem Host-PC ausgelesen werden. Durch die Ausnutzung des DSP-Speichers sind zeitlich hochaufgelöste Messungen über mehrere Sekunden Messzeit möglich.

1.5.2 Funktion des Offset-LO

Diese Arbeit behandelt die Implementierung eines Offset-LO, wie er in Abbildung 1.2 dargestellt ist. Dieser ist im Moment mit einer analogen PLL-Schaltung realisiert, die nicht mehr weiterentwickelt wird. Daher sollte ein digitales Äquivalent entwickelt werden, welches die Vorteile digitaler Schaltungen bezüglich ihrer Flexibilität, Reproduzierbarkeit und Abgleichfreiheit mitbringt. Ein Offset-LO kann von seiner äußeren Funktion einfach beschrieben werden:

Aus zwei sinusförmigen Eingangssignalen konstanter Amplitude und variabler Frequenz soll ein sinusförmiges Ausgangssignal konstanter Amplitude erzeugt werden, dessen Frequenz gleich der Summe der Frequenzen der Eingangssignale ist.

Die naheliegendste Operation die eine solche Frequenzverschiebung erzielt, liefert ein idealer Mischer. Ein idealer Mischer multipliziert zwei Signale. Für die Signale

$$s_1(t) = \cos(\omega_1 t) , \quad s_2(t) = \cos(\omega_2 t) \quad (1.8)$$

entsteht am Ausgang des Mixers ein Signal der Form

$$s_1(t) \cdot s_2(t) = \frac{1}{2} \left(\cos(\omega_1 - \omega_2)t + \cos(\omega_1 + \omega_2)t \right) . \quad (1.9)$$

Durch Herausfiltern der Summenfrequenz lässt sich somit das gewünschte Offset-LO Verhalten erzielen. Leider haben reale Mischer die Eigenschaft, auch Nichtlinearitäten höherer Ordnung zu haben. Somit entstehen prinzipiell Frequenzkomponenten bei ganzen Vielfachen $n \omega_1 \pm m \omega_2$ ($n, m \in \mathbb{Z}_0$) der Grundfrequenz. Das gleiche gilt für ideale Mischer mit nicht-sinusförmigen Eingangssignalen. Da der betrachtete Frequenzbereich eine Frequenzdynamik von mehreren Oktaven hat, kommt es in diesen Fällen zu einer Überlappung der Frequenzbereiche, die nicht herausgefiltert werden können.

Anders sieht es hingegen bei der Realisierung eines Offset-LO auf Basis einer PLL aus. Wie im nächsten Kapitel näher erläutert wird, besteht eine PLL aus einem eigenen Oszillator, der auf ein Eingangssignal mit Hilfe einer Regelschaltung synchronisiert

wird. Durch eine Manipulation des Eingangssignals lassen sich Offset-Frequenzen einstellen, die nicht die genannten Nachteile aufweisen. Details zur Verwendung einer PLL als Offset-LO finden sich in Kapitel 2.6 ab Seite 37.

1.5.3 Anforderungen an den Offset-LO

Die letzten Abschnitte haben gezeigt, welche Anforderungen sich aus den physikalischen und technischen Randbedingungen ergeben. Die für den Offset-LO relevanten Anforderungen sind in Tabelle 1.2 zusammengefasst. Da die Phasenauflösung des DSP-Systems aus strahldynamischen Gründen auf $\pm 5^\circ$ ausgelegt ist, sollte auch die Phasengenauigkeit des Offset-LO diesen Wert nicht überschreiten.

Parameter	Symbol	Wert
Frequenzbereich HF-Eingang	f_{RF}	0,8 bis 5,4 MHz
Frequenzbereich ZF-Eingang	f_{IF}	21,4 MHz
Frequenzbereich Ausgangsfrequenz	$f_o = f_{RF} + f_{IF}$	22,2 bis 26,8 MHz
Maximale Frequenzänderung der HF	$\dot{f}_{RF,max}$	68,08 MHz/s
Maximaler Phasenfehler	$\Delta\varphi_{max}$	$\leq \pm 5^\circ$

Tabelle 1.2: Anforderungen an den Offset-LO

2 Grundlagen der Phase-Locked Loop

In diesem Kapitel soll ausführlich auf die Grundlagen der Phase-Locked Loop (PLL) eingegangen werden. Diese sind für die Auslegung der PLL des Offset-LO notwendig. Dennoch ist das Kapitel allgemein gehalten und kann von Lesern mit fundierten PLL-Kenntnissen übersprungen werden. Zunächst wird auf den allgemeinen Aufbau und die Grundbegriffe der PLL eingegangen. Anschließend erfolgt die Behandlung der zeitkontinuierlichen (analogen) PLL. Obwohl eine digitale PLL implementiert wurde, ist dies notwendig, da im darauf folgenden Kapitel diese Ergebnisse verwendet werden, um die Parameter der zeitdiskreten PLL zu ermitteln.

2.1 Funktionsweise einer PLL

Abbildung 2.1 zeigt den Aufbau einer konventionellen, analogen PLL-Schaltung mit Frequenzteilung. Ein sinusförmiges Eingangssignal der Form

$$u_i(t) = \hat{u}_i \sin(\omega_i t + \phi_i) \quad (2.1)$$

wird der PLL als Referenz zugeführt und gelangt zunächst auf einen Eingangsfrequenzteiler (: N_i). Dieser erzeugt ein Signal¹

$$u'_i(t) = \hat{u}'_i \sin((\omega_i t + \phi_i)/N_i) \quad (2.2)$$

welches mit dem frequenzgeteilten Ausgangssignal des spannungsgesteuerten Oszillators (VCO)

$$u'_o(t) = \hat{u}'_o \sin((\omega_o t + \phi_o)/N_o) \quad (2.3)$$

über einen Phasendetektor (PD) in der Phase verglichen wird.

¹Je nach verwendetem Frequenzteiler können auch andere Signalformen (z.B. Rechtecksignale) entstehen.

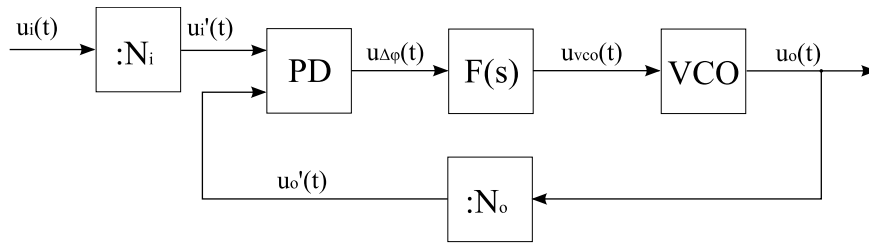


Abbildung 2.1: Aufbau einer PLL

Das Ausgangssignal eines idealen Phasendetektors liefert die Differenz der Argumente beider Sinus-Funktionen

$$u_{\Delta\varphi}(t) = K_d \Delta\varphi(t) \quad (2.4)$$

mit

$$\Delta\varphi(t) = \left(\frac{\omega_i}{N_i} - \frac{\omega_o}{N_o} \right) t + \frac{\phi_i}{N_i} - \frac{\phi_o}{N_o} \quad (2.5)$$

Man erhält zunächst einen zum Frequenzfehler proportionalen Anteil, der linear mit der Zeit ansteigt, sowie einen konstanten Phasenfehler. K_d entspricht der Phasendetektorverstärkung mit Einheit [V/rad] und variiert je nach verwendetem Phasendetektortyp. Dieses Fehlersignal gelangt auf einen Regler $F(s)$ (im PLL-Kontext auch Schleifenfilter genannt), der die Abstimmspannung und damit die Ausgangsfrequenz des VCO bestimmt. Das Ausgangssignal eines idealen VCO berechnet sich zu

$$u_o(t) = \hat{u}_o \sin((\omega_c + K_0 u_{VCO})t + \phi_o) \quad (2.6)$$

wobei ω_c der Mittenfrequenz (engl. *center frequency*) des VCO entspricht, wenn keine VCO-Spannung anliegt, K_0 entspricht der Oszillatorverstärkung mit der Einheit [rad/s/V]. Wählt man den noch nicht näher definierten Regler $F(s)$ so, dass die Phasendifferenz $\Delta\varphi$ zu Null geregelt wird, verschwinden Frequenz- und Phasendifferenz ($\frac{\omega_o}{N_o} = \frac{\omega_i}{N_i}$; $\frac{\phi_o}{N_o} = \frac{\phi_i}{N_i}$). Die Ausgangsfrequenz ergibt sich nun aus

$$\omega_o = \frac{N_o}{N_i} \omega_i \quad (2.7)$$

Somit kann ein Ausgangssignal erzeugt werden, dessen Frequenz in einem rationalen Verhältnis zur Eingangsfrequenz und in einem festen Phasenbezug zum Eingangssignal steht.

2.2 Grundbegriffe

In der Literatur haben sich einige Grundbegriffe durchgesetzt, die hier kurz erläutert werden sollen.

2.2.1 PLL-Struktur

Die verschiedenen Strukturen der PLL haben sich historisch gesehen weiterentwickelt und ihre eigenen Namen bekommen (in Anlehnung an [Bes93]):

- Linear PLL (LPLL): Eine typischerweise analoge PLL, ähnlich dem Blockschaltbild aus Abbildung 2.1. Als Phasendetektor kommt hierbei häufig ein Mischer (Sinus-Phasendetektor) zum Einsatz.
- Digital PLL (DPLL): Diese PLL ist ähnlich der LPLL, jedoch mit einem digitalen (Rechteck-)Phasendetektor. Dieser arbeitet zeitkontinuierlich und erzeugt ein pulswellenmoduliertes Ausgangssignal, welches nach Tiefpassfilterung einer kontinuierlichen Phase entspricht.
- All-digital PLL (ADPLL): Dieser PLL-Typ arbeitet ausschließlich zeitdiskret. Die Komponenten sind typischerweise digital implementiert. An Stelle des VCO wird ein *Direct Digital Synthesizer* (DDS) eingesetzt.

Natürlich gibt es auch Mischformen wie z.B. eine ADPLL, die analoge Komponenten enthält, aber trotzdem zeitdiskret arbeitet.

2.2.2 Arbeitsbereich einer PLL

Ein wichtiger Entwurfsparameter ist der Arbeitsfrequenzbereich einer PLL. Dabei gibt es unterschiedliche statische und dynamische Grenzwerte, die vom Ausgangszustand der PLL abhängen.

Solange der Regler die Phasendifferenz noch nicht ausgeregelt hat, spricht man von einer *ausgerasteten* PLL, andernfalls ist die PLL *ingerastet*. Wird bei einer *ausgerasteten* PLL die Frequenz des Eingangssignals in Bezug zur Mittenfrequenz des VCO sprunghaft verändert, so gibt es eine Differenzfrequenz bei der die PLL ohne *Phasenschlupf* (engl. *phase slip*) einrastet. *Ohne Phasenschlupf* bedeutet hierbei, dass der Phasenfehler den Bereich $-\pi$ bis π nie verlässt. Der Frequenzbereich, in dem solche Sprünge zulässig sind, wird Fangbereich $\Delta\omega_L$ (engl. *lock-in range*) genannt. Außerhalb dieser Frequenz gibt es einen Bereich, in dem die PLL zwar sicher einrastet,

der Phasenfehler aber unter Umständen mehrmals den Bereich $-\pi$ bis π durchläuft. Diesen Bereich nennt man Ziehbereich $\Delta\omega_{PI}$ (engl. *pull-in range*).

Ist die PLL *ingerastet* und wird die aktuelle Frequenz des Eingangssignals sprunghaft verändert, so bezeichnet man die Frequenz, bei der die PLL ausrastet, als Ausrastfrequenz $\Delta\omega_{PO}$ (engl. *pull-out frequency*) bzw. den entsprechenden Frequenzbereich Ausrastbereich (dynamische Stabilitätsgrenze). Wird die Frequenz des Eingangssignals hingegen langsam verändert, so wird der Bereich bis zu der Frequenz, bei der die PLL ausrastet, als Haltebereich $\Delta\omega_H$ (engl. *hold-in range*) bezeichnet (statische Stabilitätsgrenze). Außerhalb dieses Bereichs ist die PLL instabil. Demnach gilt

$$\Delta\omega_L < \Delta\omega_{PI} \quad (2.8)$$

für den ausgerasteten Anfangszustand und

$$\Delta\omega_{PO} < \Delta\omega_H \quad (2.9)$$

für den eingerasteten Anfangszustand. Darüber hinaus gilt für die meisten PLL [Bes93],[Kro03]

$$\Delta\omega_L < \Delta\omega_{PO} < \Delta\omega_{PI} < \Delta\omega_H . \quad (2.10)$$

2.3 Grundgleichungen der zeitkontinuierlichen PLL

2.3.1 Modellierung der PLL mit Hilfe von Phasensignalen

Für den Entwurf des Reglers und der Systemparameter ist eine Beschreibung im Bildbereich hilfreich. Hierbei ergibt sich zunächst das Problem, dass das System nichtlineare Elemente wie z.B. den Phasendetektor enthält. Dieses Problem kann man für den Bereich umgehen, in dem der Phasendetektor für Eingangsphasen linear (z.B. Dreieck- oder Sägezahn-Detektor) bzw. näherungsweise linear (Sinusdetektor) arbeitet. Hierzu betrachtet man nicht mehr die Amplituden der Sinussignale, sondern deren Argumente und bezeichnet diese als Phasensignale. Das Phasensignal zum PLL-Eingangssignal (2.1) lautet somit $\varphi_i(t) = \omega_i t + \phi_i$.

Der Frequenzteiler verhält sich für Phasensignale wie ein gewöhnlicher Teiler $\varphi'_i(t) = (\omega_i t + \phi_i)/N_i$. Der Phasendetektor kann als Subtrahierer modelliert werden $\Delta\varphi(t) = \varphi'_i(t) - \varphi'_o(t)$. Der Regler bleibt unverändert, da er bereits mit Phasensignalen arbeitet. Das VCO-Ausgangsphasensignal lautet $\varphi_o(t) = (\omega_c + K_0 u_{VCO})t + \phi_o$. Die Kreisfrequenz ist bekanntlich die Ableitung der Phase nach der Zeit. Leiten wir letzte Gleichung nach der Zeit ab, erhalten wir:

$$\frac{d}{dt}\varphi_o(t) = \omega_c + K_0 u_{VCO} \quad (2.11)$$

Um daraus eine Übertragungsfunktion herzuleiten, transformieren wir diese in den Laplacebereich. Mit der Beziehung $\mathcal{L}\{\frac{d}{dt}x(t)\} = sX(s)$ erhalten wir als Übertragungsfunktion des VCOs für Phasensignale

$$\frac{\Phi_o(s)}{U_{VCO}(s)} = \frac{K_0}{s} \quad (2.12)$$

Der VCO arbeitet für Phasensignale wie ein idealer Integrator mit Verstärkung K_0 . Für die Betrachtung der Dynamik ist der Wert der konstanten Mittenfrequenz irrelevant, so dass er zu null gesetzt wurde. Mit diesen Beziehungen gelangen wir zum Phasenmodell der PLL, dessen Aufbau in Abbildung 2.2 dargestellt ist.

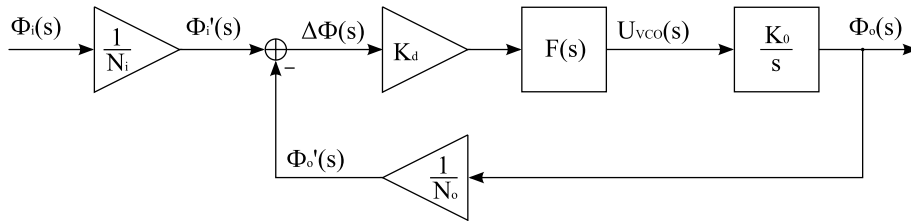


Abbildung 2.2: Phasenmodell der analogen PLL

2.3.2 Allgemeine Übertragungsfunktionen

Stellt man die Systemgleichungen dieses Blockschaltbildes auf und löst sie entsprechend auf, so erhält man die Führungsübertragungsfunktion zu

$$H(s) = \frac{\Phi_o(s)}{\Phi_i(s)} = \frac{K_0 K_d / N_i F(s)}{s + K_0 K_d / N_o F(s)} \quad (2.13)$$

sowie die Fehlerübertragungsfunktion zu

$$E(s) = \frac{\Delta\Phi(s)}{\Phi_i(s)} = \frac{1/N_i s}{s + K_d K_0 / N_o F(s)} \quad (2.14)$$

2.3.3 Ordnung und Typ einer PLL

Die Ordnung einer PLL entspricht der Ordnung der Übertragungsfunktion, d.h. die höchste Potenz des Nenners in s . Diese ist im Allgemeinen immer eins höher als

die Ordnung des Schleifenfilters $F(s)$. Der Typ hingegen gibt an, wie viele ideale Integratoren in der offenen Regelschleife vorhanden sind. Da der VCO einen Integrator enthält, ist der kleinste Typ einer PLL eins. Weitere ideale Integratoren können durch das Schleifenfilter hinzukommen und beeinflussen die stationäre Genauigkeit.

2.3.4 Übertragungsfunktionen mit Regler

In Literatur und Praxis sind die folgenden Regler bzw. Schleifenfilter weit verbreitet und sind darüber hinaus in vielen Veröffentlichungen [Bes93],[Kro03],[Gar79] ausführlich untersucht worden:

P-Regler (Proportional-Regler) :

$$F_P(s) = K_P \quad (2.15)$$

PI-Regler (Proportional/Integral-Regler) :

$$F_{PI}(s) = K_P + \frac{K_I}{s} \quad (2.16)$$

RC-Filter (Tiefpass 1. Ordnung):

$$F_{RC}(s) = \frac{K}{s\tau_1 + 1} \quad (2.17)$$

Lag-lead Filter:

$$F_{LL}(s) = K \frac{s\tau_1 + 1}{s\tau_2 + 1} \quad (2.18)$$

Mit einem P-Regler (nullte Ordnung) entsteht eine Typ I PLL erster Ordnung. Beim RC- und Lag-lead Filter resultiert eine Typ I PLL zweiter Ordnung. Nur beim PI-Regler ergibt sich eine Typ II PLL zweiter Ordnung.

Der Typ einer PLL ist generell wesentlich wichtiger als die Ordnung, da dieser direkt die stationäre Genauigkeit beeinflusst (siehe hierzu das kommende Kapitel 2.3.7 ab Seite 19), während die Ordnung nur einen Einfluss auf den Frequenzgang hat. Mit Erhöhung des Typs und der Ordnung wird das System jedoch zunehmend instabiler, folglich sind in der Literatur meistens die PLL-Typen I bis III anzutreffen.

Im Folgenden beschränken wir uns auf die Typ II PLL zweiter Ordnung, da diese, wie später gezeigt wird, bei geringem Aufwand gute Ergebnisse zeigt.

Setzt man $F_{PI}(s)$ in (2.13) ein, so erhält man

$$H_{PI}(s) = \frac{K_0 K_d K_P / N_i s + K_0 K_d K_I / N_i}{s^2 + K_0 K_d K_P / N_o s + K_0 K_d K_I / N_o} \quad (2.19)$$

für die Führungsübertragungsfunktion, sowie

$$E_{PI}(s) = \frac{s^2 / N_i}{s^2 + K_0 K_d K_P / N_o s + K_0 K_d K_I / N_o} \quad (2.20)$$

für die Fehlerübertragungsfunktion.

Systeme zweiter Ordnung lassen sich anschaulich durch die beiden Parameter Eigenfrequenz ω_n (engl. *natural frequency*) und Dämpfung ζ (engl. *damping ratio*) darstellen, wobei der Nenner der Übertragungsfunktion (charakteristische Funktion) die Form

$$N(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (2.21)$$

aufweist. Dies gelingt im behandelten Beispiel durch die Substitutionen

$$\omega_n = \sqrt{K_0 K_d K_I / N_o} \quad (2.22)$$

und

$$\zeta = \frac{K_P}{2} \sqrt{\frac{K_0 K_d}{K_I N_o}}. \quad (2.23)$$

Das System kann mit Hilfe dieser zwei Parameter spezifiziert werden. Die benötigten Reglerkonstanten erhält man nach Umstellen der letzten beiden Gleichungen zu

$$K_P = \frac{2\zeta\omega_n N_o}{K_0 K_d} \quad (2.24)$$

und

$$K_I = \frac{\omega_n^2 N_o}{K_0 K_d}. \quad (2.25)$$

Die resultierende Führungsübertragungsfunktion lautet hiermit

$$H_{PI}(s) = \frac{N_o}{N_i} \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.26)$$

und die Fehlerübertragungsfunktion entsprechend

$$E_{PI}(s) = \frac{1}{N_i} \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \quad (2.27)$$

2.3.5 Stabilität

Ist die Lage aller Polstellen eines Systems bekannt, lässt sich sehr einfach die Bedingung für die Stabilität angeben:

Ein System $H(s)$ ist genau dann stabil, wenn sämtliche Pole s_∞ auf der linken s -Halbebene liegen ($\text{Re}\{s_\infty\} < 0$). [Unb05]

Bei der PLL mit PI-Regler nach (2.26) ergeben sich die Polstellen nach Nullsetzen des Zählerpolynoms zu

$$s_{\infty 1/2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \quad . \quad (2.28)$$

Der Realteil ist negativ für alle $\zeta > 0$, d.h. für $\zeta \leq 0$ zeigt die PLL instabiles Verhalten.

Bei Systemen zweiter Ordnung lassen sich die Polstellen einfach angeben. Bei Systemen höherer Ordnung lassen sich diese oft nur numerisch ermitteln, so dass dies nur für explizit bekannte Reglerparameter erfolgen kann. Ein weiteres beliebtes Kriterium zur Bestimmung der Stabilität ist das vereinfachte Nyquistkriterium. Es besagt, dass ein System $H(s)$ genau dann stabil ist, wenn an der Stelle $|H(j\omega_D)| = 0\text{dB}$ die Phase $\angle H(j\omega_D) > -180^\circ$ ist. Der Abstand zur -180° Linie im Phasengang liefert die Phasenreserve

$$\varphi_R = \angle H(j\omega_D) + 180^\circ \quad (2.29)$$

welche in einem stabilen System stets größer Null sein muss [Unb05].

2.3.6 Bodediagramm

Neben dem damit verbundenen Stabilitätskriterium liefert das Bodediagramm noch weitere Informationen über das Verhalten eines Systems. Abbildung 2.3 zeigt das Bodediagramm der Führungsübertragungsfunktion für verschiedene Dämpfungen. Die Frequenzachse wurde auf die Eigenfrequenz der PLL normiert. Man erkennt, dass die PLL Tiefpassverhalten für Phasenmodulationen (ω_m) des Eingangssignals aufweist. Bis zur Eigenfrequenz kann die Ausgangsphase der PLL der Eingangsphase folgen, höherfrequente Anteile werden unterdrückt. Für Dämpfungswerte $\zeta < 1/\sqrt{2} \approx 0,707$ ist eine deutliche Resonanzüberhöhung erkennbar, große Dämpfungswerte führen hingegen zu einem weniger ausgeprägtem Tiefpassverhalten. Im Gegensatz dazu zeigt das Bodediagramm der Fehlerübertragungsfunktion Hochpassverhalten (Abbildung 2.4). Niederfrequente Phasenmodulationsanteile können von der PLL gut verfolgt werden. Bei Frequenzanteilen, die über der Eigenfrequenz liegen, entsteht jedoch ein Phasenfehler von fast 100%, da das Eingangssignal praktisch ungedämpft auf den Ausgang wirkt.

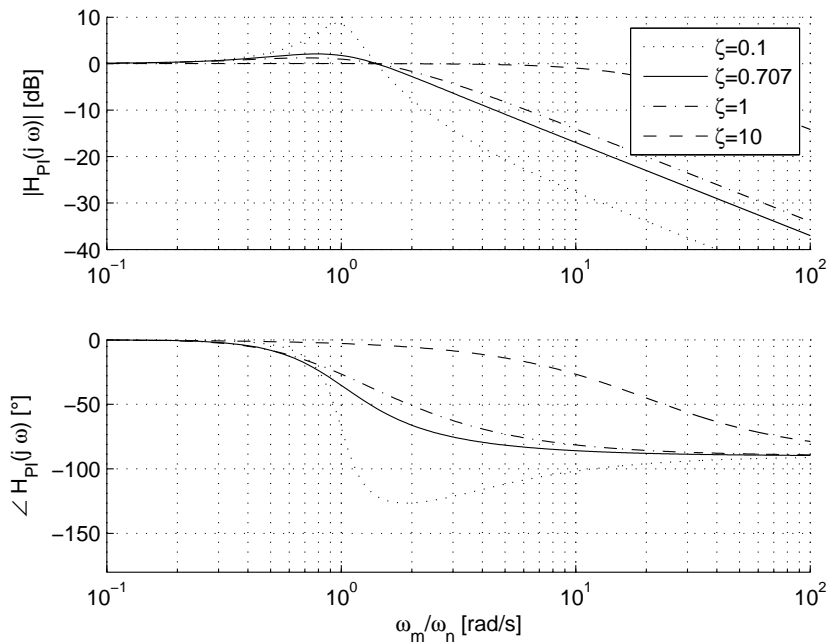


Abbildung 2.3: Bodediagramm der Führungsübertragungsfunktion

Eine Phasenmodulation kann durch Integration in eine Frequenzmodulation umgewandelt werden. Dabei wird die Modulationsfrequenz nicht verändert. Daher kann die PLL auch als schmalbandiges, adaptives Bandpassfilter aufgefasst werden. Die PLL synchronisiert sich auf die Frequenzkomponente des Eingangssignals mit der größten Leistung. Die Frequenz der eingerasteten PLL entspricht somit der adaptiven Mittenfrequenz des Bandpassfilters. Um diese Mittenfrequenz werden nach Abbildung 2.3 am Ausgang der PLL Frequenzanteile durchgelassen, die unterhalb der Eigenfrequenz ω_n der PLL liegen. Dabei kann die Eigenfrequenz sehr klein gegenüber den Signalfrequenzen sein.

2.3.7 Stationäre Genauigkeit

Ein wichtiges Kriterium für den Reglerentwurf ist der Regelfehler, der sich im stationären Zustand einstellt. Dieser entspricht $\Delta\phi$ für $t \rightarrow \infty$ welches im Folgenden mit $\Delta\phi_\infty$ abgekürzt wird. Über den Endwertsatz der Laplacetransformation [Fli91]

$$\lim_{t \rightarrow \infty} g(t) = \lim_{s \rightarrow 0} sG(s) \quad (2.30)$$

lässt sich dieser einfach im Bildbereich ermitteln. Der Regelfehler ist abhängig von der Form des Eingangssignals. Prinzipiell kann hier jedes beliebige Eingangssignal

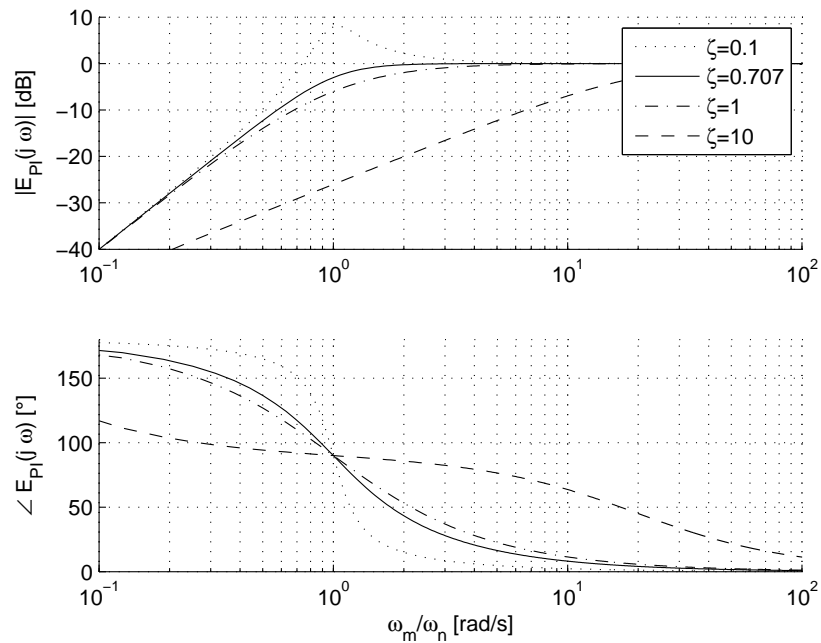


Abbildung 2.4: Bodediagramm der Fehlerübertragungsfunktion

herangezogen werden, wobei im PLL-Kontext einige Eingangssignale für die Analyse besonders hilfreich sind.

Angenommen nach dem Einschalten der PLL schwingt das Eingangssignal bei einer Frequenz ω mit einer Nullphase ϕ_i . Das entsprechende Phasensignal lautet $\varphi_i(t) = \omega t + \phi_i$. Angenommen das Ausgangssignal (VCO) schwingt bei gleicher Frequenz ω , jedoch mit einem Phasenoffset von $\Delta\phi = \phi_i - \phi_o$. Dieser Fall erzeugt das gleiche Ergebnis, als würde man bei einer eingeschwingenen PLL einen Phasensprung $\Delta\phi \epsilon(t)$ auf den Eingang geben. Mit $\epsilon(t)$ ist hierbei die Sprungfunktion

$$\epsilon(t) = \begin{cases} 1 & \text{für } t > 0 \\ 0 & \text{für } t \leq 0 \end{cases} \quad (2.31)$$

bezeichnet.

Analog dazu lässt sich ein konstanter Frequenzunterschied $\Delta\omega$ beim Einschalten durch einen Frequenzsprung modellieren, welcher einer Phasenrampe $\Delta\omega t \epsilon(t)$ entspricht. Ein lineares Ändern der Frequenz (Frequenzrampe) führt letztendlich zu einer quadratisch von der Zeit abhängenden Phasenfunktion. Tabelle 2.1 fasst die eben genannten Eingangssignalformen zusammen und gibt deren Laplacetransformierte wieder.

Eingangssignal	$\varphi(t)$	$\Phi(s)$
Phasensprung	$\Delta\phi \epsilon(t)$	$\Delta\phi \frac{1}{s}$
Frequenzsprung	$\Delta\omega t \epsilon(t)$	$\Delta\omega \frac{1}{s^2}$
Frequenzrampe	$\Delta\dot{\omega} t^2 \epsilon(t)$	$\Delta\dot{\omega} \frac{2}{s^3}$

Tabelle 2.1: Kontinuierliche Eingangssignalformen [IB05]

Der Frequenzfehler im Bildbereich ist über (2.14) gegeben

$$\Delta\Phi(s) = \Phi_i(s)E(s) \quad (2.32)$$

was zu einem stationärem Phasenfehler von

$$\Delta\varphi_\infty = \lim_{t \rightarrow \infty} \Delta\varphi(t) = \lim_{s \rightarrow 0} s\Delta\Phi(s) = \Phi_i(s) \frac{1/N_i s^2}{s + K_d K_0/N_o F(s)} \quad (2.33)$$

führt. Im Falle des Phasensprungs ergibt sich ein stationärer Phasenfehler von

$$\Delta\varphi_\infty = \lim_{s \rightarrow 0} \Delta\varphi \frac{1/N_i s}{s + K_d K_0/N_o F(s)} = 0 \quad , \quad (2.34)$$

solange der Nenner ungleich Null ist. Dies lässt sich erfüllen, solange der Regler nicht ausschließlich aus differentiellen Anteilen besteht. Im Falle des Frequenzsprungs ergibt sich

$$\Delta\varphi_\infty = \lim_{s \rightarrow 0} \Delta\omega \frac{1/N_i}{s + K_d K_0/N_o F(s)} \quad . \quad (2.35)$$

Der stationäre Phasenfehler kann genau dann zu Null geregelt werden, wenn der Regler die Bedingung

$$\lim_{s \rightarrow 0} F(s) = \infty \quad (2.36)$$

erfüllt, d.h. eine Polstelle für $s=0$ hat und somit mindestens ein I-Glied enthält. Bei der Frequenzrampe ergibt sich analog

$$\Delta\varphi_\infty = \lim_{s \rightarrow 0} \Delta\dot{\omega} \frac{2}{s} \frac{1/N_i}{s + K_d K_0/N_o F(s)} \quad . \quad (2.37)$$

welches die Bedingung

$$\lim_{s \rightarrow 0} sF(s) = \infty \quad (2.38)$$

erfordert, um den stationäre Phasenfehler komplett auszuregeln. Dies ist nur mit minimal zwei I-Gliedern möglich (z.B. $F(s) = 1/s^2$). Mit einem PI-Regler nach (2.16) ergibt sich zusammen mit (2.25) ein bleibender Fehler von

$$\Delta\varphi_\infty = \frac{2\dot{\Delta}\omega}{K_I K_d K_0 N_i / N_o} = \frac{2\dot{\Delta}\omega}{\omega_n^2 N_i} . \quad (2.39)$$

Durch eine hohe Verstärkung des I-Anteils (K_I) bzw. durch eine hohe Eigenfrequenz, kann der stationäre Phasenfehler entsprechend reduziert werden.

2.3.8 Zeitverhalten

Neben dem stationären Fehler ist der zeitliche Verlauf für verschiedene Eingangssignale interessant. Abbildung 2.5(a) zeigt die Ausgangsphase der PLL mit PI-Regler nach einem Phasensprung der Eingangsphase um 1° für verschiedene Dämpfungen. Abbildung 2.5(b) zeigt die Sprungantwort der Fehlerübertragungsfunktion. Die Zeitachse wurde auf die Periodenlänge der Eigenfrequenz ($T_n = 2\pi/\omega_n$) normiert.

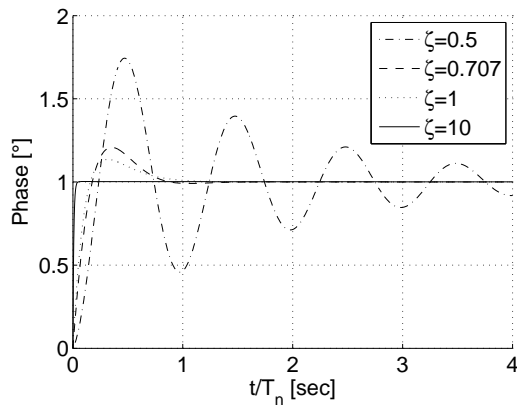
Ein weiteres wichtiges Eingangssignal ist der Frequenzsprung, der einer rampenförmigen Phase entspricht. In Abbildung 2.5(c) ist die Antwort des Phasensignals auf ein Frequenzsprung von ω_n dargestellt. Interessanter ist hingegen die Antwort des Phasenfehlers in Abbildung 2.5(d). Der Frequenzfehler kann offensichtlich mit dieser PLL ausgeregelt werden.

Markant ist das hohe Überschwingen für kleine Dämpfungswerte. Für Dämpfungswerte nahe Eins ($\zeta = 0,707$ bis 2) ist das System nach max. $2T_n$ eingeschwungen. Für hohe Dämpfungswerte zeigt die Phasen-Sprungantwort zwar ein kurzes Einschwingverhalten (Abb. 2.5(b), $\zeta = 10$), das Einschwingen des Phasenfehlers bei einem Frequenzsprung dauert dafür um so länger (Abb. 2.5(d), $\zeta = 10$). Daher sind oft Dämpfungswerte zwischen 0.5 und 2 in der Praxis anzutreffen [Gar05].

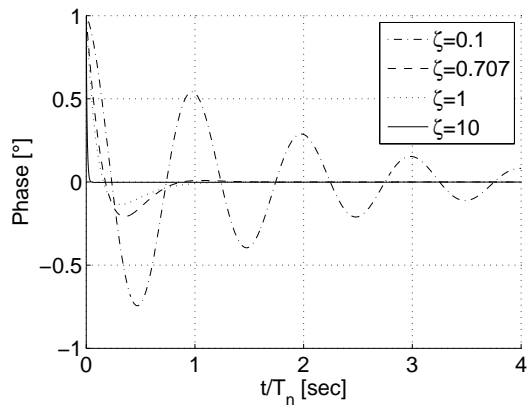
2.4 Grundgleichungen der zeitdiskreten PLL

2.4.1 Berechnung der z-Übertragungsfunktion als Näherung für kontinuierliche Systeme

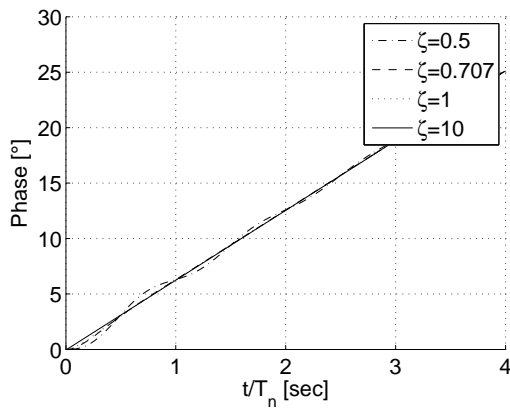
Diskrete Signale und Systeme lassen sich vorteilhaft mit Hilfe der z-Transformation beschreiben [Fli91]. Liegt ein diskretes System z.B. in Form einer Differenzgleichung vor, so lässt sich die z-Transformierte direkt durch Anwendung der Transformationsbeziehung angeben. Möchte man hingegen ein kontinuierliches System durch ein äquivalentes diskretes System beschreiben, so ist dies nur für bestimmte Eingangssignale



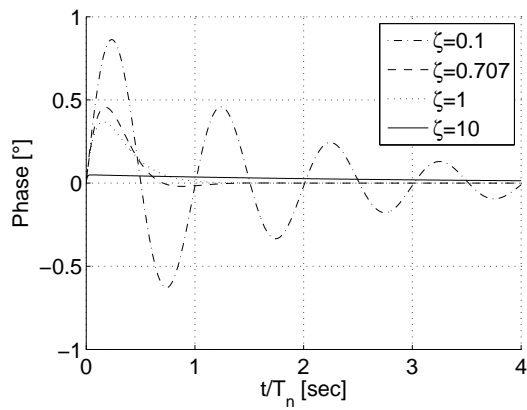
(a) Phasensprungantwort $H(s)$



(b) Phasensprungantwort $E(s)$



(c) Frequenzsprungantwort $H(s)$



(d) Frequenzsprungantwort $E(s)$

Abbildung 2.5: Phasen- und Frequenz-Sprungantworten von $H(s)$ und $E(s)$

exakt möglich [Unb00]. Die formale Beziehung zwischen der Laplacetransformierten eines diskreten Systems und der z-Transformierten lautet

$$z = e^{sT_s} \quad (2.40)$$

bzw.

$$s = \frac{1}{T_s} \ln z . \quad (2.41)$$

Die einfachste Methode zur Diskretisierung einer Differentialgleichung ist das Euler-Verfahren, bei dem der Differentialquotient durch einen Differenzenquotienten mit kleiner Schrittweite T_s approximiert wird:

$$\left. \frac{df(t)}{dt} \right|_{t=nT_s} \approx \frac{1}{T_s} (f(nT_s) - f((n-1)T_s)); \quad n \in \mathbb{Z}_0 \quad (2.42)$$

Eine Differentialgleichung der Form

$$y(t) = \frac{d}{dt} x(t) , \quad (2.43)$$

welche im Bildbereich

$$Y(s) = sX(s) \quad (2.44)$$

lautet, kann so durch die Differenzengleichung

$$y(n) = \frac{1}{T_s} (x(nT) - x((n-1)T_s)) \quad (2.45)$$

angenähert werden. Die Anwendung der z-Transformation liefert

$$Y(z) = \frac{1}{T_s} (1 - z^{-1}) X(z) . \quad (2.46)$$

Ein Vergleich mit (2.44) liefert die Korrespondenz [Unb00]

$$s \leftrightarrow \frac{1}{T_s} (1 - z^{-1}) . \quad (2.47)$$

Eine einfache Approximation eines kontinuierlichen Systems $H(s)$ kann somit durch Einsetzen der Substitutionsbeziehung

$$s = \frac{1}{T_s} (1 - z^{-1}) = \frac{z - 1}{T_s z} \quad (2.48)$$

in die kontinuierliche Übertragungsfunktion erzielt werden. Diese Beziehung wird auch Rechteckregel genannt, da die kontinuierlichen Signale durch Treppenfunktionen dargestellt werden. Sie ist jedoch nur für kleine Abtastzeiten ausreichend genau.

Eine bessere Approximationsbeziehung erhält man aus (2.41), indem man die Logarithmusfunktion durch eine Reihenentwicklung darstellt:

$$s = \frac{1}{T_s} \ln z = \frac{1}{T_s} 2 \left[\frac{z-1}{z+1} + \frac{1}{3} \left(\frac{z-1}{z+1} \right)^3 + \frac{1}{5} \left(\frac{z-1}{z+1} \right)^5 + \dots \right] \quad (2.49)$$

Durch Abbruch nach dem ersten Glied erhält man die sogenannte Tustin-Formel

$$s \approx \frac{2}{T_s} \frac{z-1}{z+1}, \quad (2.50)$$

welche auch Trapezregel oder bilineare Transformation genannt wird. Sie hat die Eigenschaft, die linke Seite der s-Halbebene ($Re\{s\} < 0$) in das Innere des Einheitskreises der z-Ebene ($|z| < 1$) zu transformieren, womit die Stabilitätseigenschaften erhalten bleiben.

Allerdings ist diese aufwendiger, und es entstehen in der Regel zeitdiskrete Systeme höherer Ordnung als mit der Rechteckregel. Daher muss je nach Anwendung abgewägt werden, ob der Aufwand gerechtfertigt ist.

2.4.2 Direkte digitale Frequenzsynthese

Anstelle eines VCO kommen bei der digitalen PLL, Verfahren der direkten digitalen Synthese (engl. *Direct Digital Synthesizer*, DDS) zum Einsatz. Oft werden diese auch als *Number Controlled Oscillator* (NCO) in Anlehnung an den VCO bezeichnet. Abbildung 2.6 zeigt den Aufbau eines DDS. Über die beiden Steuerparameter *Frequency Tuning Word* (FTW) und *Phase Tuning Word* (PTW) können Frequenz und Phase des Synthesizers eingestellt werden. Beide Parameter sind ganze Zahlen mit einem Wertebereich von 0 bis $2^M - 1$.

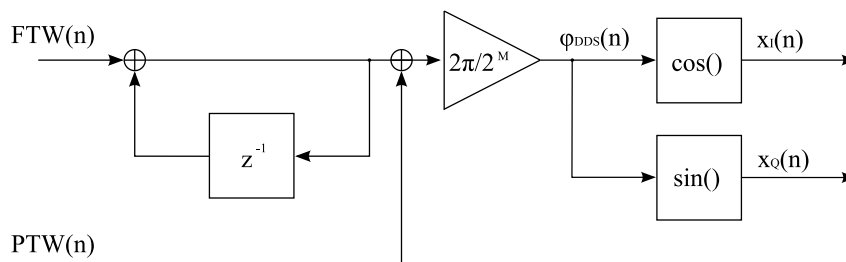


Abbildung 2.6: Aufbau eines DDS

Die Addierer arbeiten modulo 2^M . Das Signal am Ausgang des zweiten Addierers wird auf 2π normiert². Für konstantes FTW und PTW entsteht das Phasensignal

²In der praktischen Implementierung geschieht dies in der LUT.

$$\varphi_{DDS}(n) = \frac{2\pi}{2^M}((nFTW + PTW) \bmod 2^M) . \quad (2.51)$$

Aus diesem Phasensignal wird direkt mit einer Cosinus-/Sinus *Look-Up-Table* (LUT) ein In-Phase- und ein (optionales) Quadratur-Phase³ Signal erzeugt:

$$x_i(n) = \cos\left(\frac{2\pi}{2^M}(nFTW + PTW)\right) \quad (2.52)$$

$$x_q(n) = \sin\left(\frac{2\pi}{2^M}(nFTW + PTW)\right) \quad (2.53)$$

Dies entspricht der diskreten Wertefolge einer abgetasteten Zeitfunktion $x_i(t)$ zum Zeitpunkt $t = nT_s$

$$x_i(t = nT_s) = \cos\left(\underbrace{\frac{2\pi}{2^M} \frac{1}{T_s} FTW t}_{=\omega_{DDS}} + \underbrace{\frac{2\pi}{2^M} PTW}_{\phi_{DDS}}\right) . \quad (2.54)$$

Die Ausgangsfrequenz eines DDS lautet somit mit $f_s = 1/T_s$

$$f_{DDS} = \frac{FTW}{2^M} f_s \quad (2.55)$$

und die Anfangsphase

$$\phi_{DDS} = \frac{PTW}{2^M} 2\pi . \quad (2.56)$$

Die Frequenzauflösung beträgt somit

$$\Delta f_{DDS} = \frac{f_s}{2^M} . \quad (2.57)$$

Man sieht, dass für eine feine Frequenzauflösung eine hohe FTW-Wortbreite nötig ist. Bei kommerziellen DDS-Halbleitern beträgt diese bis zu 48 Bit [Ana06]. Da der Speicherbedarf einer LUT mit 2^M steigt (mit jedem Bit Frequenzauflösung verdoppelt sich die LUT-Größe), werden üblicherweise nur die höherwertigen Bit des Phasenakkumulators für die LUT ausgewertet. Dieses Abschneiden der Phase (im Englischen als *phase truncation* bezeichnet) führt zu einer ungewollten, sägezahnförmigen Phasenmodulation, welche sich in Form von Nebenlinien im Spektrum (engl. *spurious lines*) bemerkbar machen. Dieses Thema wurde zuerst in [Meh83] untersucht, und später in [NS87] weiter präzisiert. Außerdem sind in [NS87] Abschätzungen zur maximalen Leistung der Nebenlinien in Bezug auf die Leistung

³In-Phase entspricht dem Cosinus-Ausgang, Quadratur-Phase dem Sinus-Ausgang

des Nutzsignals (engl. *spurious free dynamic range*, SFDR) und in Abhängigkeit der LUT-Eingangswortbreite angegeben. Tabelle 2.2 gibt die Resultate für eine Amplitudenaufösung von 12 Bit wieder. In [NSK88] und [NS91] werden Methoden zur effizienteren Speicherung der LUT vorgestellt. In [OM91] werden Maßnahmen vorgestellt, die Leistung der diskreten Nebenlinien durch *noise shaping* über den gesamten Frequenzbereich zu verteilen, um bessere SFDR-Werte zu erzielen. Die in diesem Abschnitt genannten Artikel wurden in [Kro98] zusammengefasst und nachgedruckt.

Phasenwortbreite	SFDR
8	53 dBc
10	58 dBc
12	68 dBc
14	74 dBc
16	77 dBc

Tabelle 2.2: SFDR-Werte einer DDS in Abhängigkeit von der Phasenaufösung [NS87]

Die Ausgangsbitbreite der LUT kann unabhängig von der Eingangsbitbreite der LUT gewählt werden. Durch die unvermeidliche Quantisierung des Amplitudensignals entstehen ebenfalls Störsignale in Form von näherungsweise weißem Quantisierungsrauschen. Das Signal-Rausch-Verhältnis (SNR) eines mit N Bit dargestellten Amplitudensignals beträgt [OS99]

$$SNR = N \cdot 6dB + 1,76dB . \quad (2.58)$$

Es muss daher eine sorgfältige Wahl der Wortbreiten bei der Auslegung eines DDS getroffen werden.

Betrachtet man den DDS im Bereich der Phasensignale zwischen $-\pi$ und π , so kann eine Übertragungsfunktion für das in diesem Bereich lineare System angegeben werden. Für eine Anfangsphase von Null (PTW=0) lautet diese

$$H_{DDS}(z) = \frac{\Phi_{DDS}(z)}{FTW(z)} = \frac{k_0}{1 - z^{-1}} = \frac{k_0 z}{z - 1} \quad (2.59)$$

mit $k_0 = 2\pi/2^M f_s$. Sie entspricht in der Struktur einem durch die Rechteckregel (2.48) approximierten Integrator.

2.4.3 Übertragungsfunktion der zeitdiskreten PLL

In Abbildung 2.7 ist das Phasenmodell der zeitdiskreten (AD-)PLL gezeigt. Der wesentliche Unterschied im Vergleich zum Blockschaltbild der zeitkontinuierlichen PLL

ist, dass hier Verzögerungszeiten (Totzeiten) berücksichtigt werden können. Diese entstehen z.B. durch Verarbeitungszeiten der diskreten Algorithmen und können in ganzen Takten als D_{FF} (engl. *feedforward delay*) und D_{FB} (engl. *feedback delay*) angegeben werden.

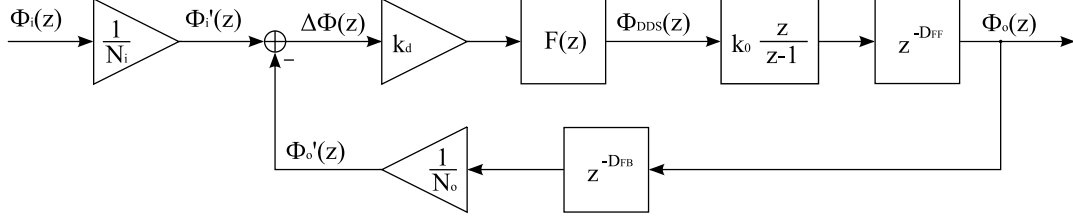


Abbildung 2.7: Phasenmodell der zeitdiskreten PLL

Stellt man die Systemgleichungen dieses Blockschaltbildes auf und löst sie entsprechend auf, so erhält man die Führungsübertragungsfunktion zu

$$H(z) = \frac{\Phi_o(z)}{\Phi_i(z)} = \frac{k_0 k_d / N_i F(z) z^{-D_{FF}}}{1 - z^{-1} + k_0 k_d / N_o F(z) z^{-(D_{FB} + D_{FF})}} \quad , \quad (2.60)$$

sowie die Fehlerübertragungsfunktion zu

$$E(z) = \frac{\Delta\Phi(z)}{\Phi_i(z)} = \frac{1/N_i (1 - z^{-1})}{1 - z^{-1} + k_d k_0 / N_o F(z) z^{-(D_{FB} + D_{FF})}} \quad . \quad (2.61)$$

Zum Vergleich liefert die Anwendung der Rechteckregel (2.48) auf das kontinuierliche System (2.13)

$$\tilde{H}(z) = H(s = \frac{1}{T_s}(1 - z^{-1})) = \frac{K_0 K_d T_s / N_i F(z)}{1 - z^{-1} + K_0 K_d T_s / N_o F(z)} \quad . \quad (2.62)$$

Die diskrete Übertragungsfunktion $H(z)$ entspricht exakt der mit der Rechteckregel transformierten, kontinuierlichen Übertragungsfunktion (2.13), für den Fall $D_{FF} = D_{FB} = 0$ und $k_0 k_d = K_0 K_d T_s$.

2.4.4 Übertragungsfunktion mit Regler

Die Anwendung der Rechteckregel (2.48) auf das analoge PI-Glied (2.16) liefert die Übertragungsfunktion

$$F_{PI}(z) \approx F_{PI}(s = \frac{1}{T_s}(1 - z^{-1})) = \frac{K_P + K_I T_s - K_P z^{-1}}{1 - z^{-1}} = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \quad , \quad (2.63)$$

mit den Koeffizienten

$$a_1 = -1 \quad (2.64)$$

$$b_0 = K_P + K_I T_s \quad (2.65)$$

$$b_1 = -K_P . \quad (2.66)$$

Die Führungs- und Fehlerübertragungsfunktionen der diskreten PLL mit PI-Regler (approximiert nach der Rechteckregel) lauten somit

$$H_{PI}(z) = \frac{K_0 K_d T_s (K_P + K_I T_s) / N_i z^{-D_{FF}} - K_0 K_d K_P T_s / N_i z^{-(D_{FF}+1)}}{z^{-2} - 2z^{-1} + 1 + K_0 K_d T_s / N_o ((K_P + K_I T_s) z^{-(D_{FF}+D_{FB})} - K_P z^{-(D_{FF}+D_{FB}+1)})} \quad (2.67)$$

und

$$E_{PI}(z) = \frac{1/N_i (z^{-2} - 2z^{-1} + 1)}{z^{-2} - 2z^{-1} + 1 + K_0 K_d T_s / N_o ((K_P + K_I T_s) z^{-(D_{FF}+D_{FB})} - K_P z^{-(D_{FF}+D_{FB}+1)})} . \quad (2.68)$$

Für den speziellen Fall $D_{FF} = D_{FB} = 0$ und $k_0 k_d = K_0 K_d T_s$ entsprechen die Übertragungsfunktionen (2.67) und (2.68) wieder exakt einer Diskretisierung der kontinuierlichen Übertragungsfunktionen (2.19) und (2.20) nach der Rechteckregel

$$\tilde{H}_{PI}(z) = \frac{K_0 K_d T_s (K_P + K_I T_s) / N_i - K_0 K_d K_P T_s / N_i z^{-1}}{z^{-2} - (K_0 K_d K_P T_s / N_o + 2) z^{-1} + K_0 K_d T_s (K_P + K_I T_s) / N_o + 1} \quad (2.69)$$

und

$$\tilde{E}_{PI}(z) = \frac{1/N_i (z^{-2} - 2z^{-1} + 1)}{z^{-2} - (K_0 K_d K_P T_s / N_o + 2) z^{-1} + K_0 K_d T_s (K_P + K_I T_s) / N_o + 1} . \quad (2.70)$$

Legt man wie beim kontinuierlichen Regler die Entwurfsparameter Eigenfrequenz ω_n und Dämpfung ζ zugrunde, lassen sich für diesen Fall die Koeffizienten aus (2.24) und (2.25) ermitteln. Daraus ergeben sich die Koeffizienten des diskreten Reglers zu

$$a_1 = -1 \quad (2.71)$$

$$b_0 = \frac{(2\zeta + \omega_n T_s) \omega_n N_o}{K_0 K_d} \quad (2.72)$$

$$b_1 = -\frac{2\zeta \omega_n N_o}{K_0 K_d} . \quad (2.73)$$

Somit ist es gelungen, die 5 bzw. 6 Koeffizienten der diskreten Übertragungsfunktionen (2.69) und (2.70), auf zwei aussagekräftige Entwurfsparameter zu reduzieren. Wie später noch gezeigt wird, ist dieser Ansatz näherungsweise auch für zeitdiskrete PLLs mit Totzeiten ($D_{FF} > 0$, bzw. $D_{FB} > 0$) anwendbar.

2.4.5 Stationäre Genauigkeit

Die Ermittlung der stationären Genauigkeit bei der zeitdiskreten PLL erfolgt analog zur Vorgehensweise wie bei der zeitkontinuierlichen PLL, nur dass diese im z -Bereich erfolgt. Der Endwertsatz der z -Transformation [Fli91] liefert

$$\lim_{n \rightarrow \infty} g(n) = \lim_{z \rightarrow 1} (z - 1)G(z) \quad . \quad (2.74)$$

Die z -Transformierten der Eingangssignale sind in Tabelle 2.3 angegeben. Der stationäre Fehler ergibt sich aus der zeitdiskreten Übertragungsfunktion (2.61) zu

$$\Delta\varphi_\infty = \lim_{z \rightarrow 1} (z - 1)\Phi_i(z) \frac{1/N_i (1 - z^{-1})}{1 - z^{-1} + k_d k_0 / N_o F(z) z^{-(D_{FB} + D_{FF})}} \quad . \quad (2.75)$$

Für den Phasensprung ergibt sich

$$\Delta\varphi_\infty = \lim_{z \rightarrow 1} \Delta\phi z \frac{1/N_i (1 - z^{-1})}{1 - z^{-1} + k_d k_0 / N_o F(z) z^{-(D_{FB} + D_{FF})}} = 0 \quad (2.76)$$

für

$$\lim_{z \rightarrow 1} F(z) \neq 0 \quad , \quad (2.77)$$

d.h. für Regler, die nicht ausschließlich aus zeitdiskreten D-Anteilen bestehen (vgl. (2.48) oder (2.50)).

Eingangssignal	$\varphi(n)$	$\Phi(z)$
Phasensprung	$\Delta\phi \epsilon(n)$	$\Delta\phi \frac{z}{z - 1}$
Frequenzsprung	$\Delta\omega n \epsilon(n)$	$\Delta\omega T_s \frac{z}{(z - 1)^2}$
Frequenzrampe	$\Delta\dot{\omega} n^2 \epsilon(n)$	$\Delta\dot{\omega} T_s^2 \frac{z(z + 1)}{(z - 1)^3}$

Tabelle 2.3: Diskrete Eingangssignalformen [Unb00]

Für den Frequenzsprung ergibt sich

$$\Delta\varphi_\infty = \lim_{z \rightarrow 1} \Delta\omega \frac{T_s / N_i}{1 - z^{-1} + k_d k_0 / N_o F(z) z^{-(D_{FB} + D_{FF})}} \quad . \quad (2.78)$$

Der Phasenfehler ist nur dann Null wenn

$$\lim_{z \rightarrow 1} F(z) = \infty \quad (2.79)$$

gilt. Dies ist für Regler mit mindestens einem diskreten Integrator der Fall.

Bei der Frequenzrampe ergibt sich ebenfalls analog zur zeitkontinuierlichen PLL, dass ein Regler mit zwei diskreten Integratoren den Phasenfehler zu Null regelt. Wird ein PI-Regler nach (2.63) verwendet, ist der resultierende Phasenfehler

$$\Delta\varphi_\infty = \frac{2\dot{\Delta}\omega}{K_I K_d K_0 N_i / N_o} = \frac{2\dot{\Delta}\omega}{\omega_n^2 N_i} . \quad (2.80)$$

Der bleibende Phasenfehler stimmt somit exakt mit dem der zeitkontinuierlichen PLL (2.39) überein. Damit lässt sich eine minimale Eigenfrequenz aus den Anforderungen bestimmen, die für beide PLL-Typen gültig ist:

$$\omega_n = \sqrt{\frac{2\dot{\Delta}\omega}{\Delta\varphi_\infty N_i}} \quad (2.81)$$

Mit dem Eingangsteiler $N_i = 1$ ergibt sich aus den Anforderungen aus Abschnitt 1.5.3 (Seite 9) eine minimale Eigenfrequenz von

$$f_{n,min} = 15,8kHz . \quad (2.82)$$

2.4.6 Zeitverhalten

Ausschlaggebend für Abweichungen zwischen der zeitdiskreten und der zeitkontinuierlichen PLL, ist die Abtastfrequenz sowie die Verzögerungszeiten D_{FF} und D_{FB} . Bei konstanter Summenverzögerung $D = D_{FB} + D_{FF}$, ergibt sich prinzipiell die gleiche Dynamik. Wie im Phasenmodell der zeitdiskreten PLL zu sehen ist (Abbildung 2.7), wird bei $D_{FF} > 0$ und $D = const.$ lediglich der Ausgangswert verzögert. Daher wird im Folgenden $D_{FF} = 0$ gesetzt und das zeitliche Verhalten für verschiedene Werte von D ermittelt. Die Auswertung des Zeitverhaltens der Übertragungsfunktionen (2.68) und (2.67), zeigten identische Approximationseigenschaften, solange D proportional zum Verhältnis f_s/f_n verändert wird. In Abbildung 2.8 sind die Phasen- und Frequenzsprungantworten für $\Delta\varphi = 1^\circ$ und $\Delta f = 1$ Hz für unterschiedliche Verhältnisse von D zu f_s/f_n bei $\zeta = 0,707$ dargestellt. Es stellt sich heraus, dass für Werte

$$D \leq \frac{1}{100} \frac{f_s}{f_n} \quad (2.83)$$

das Zeitverhalten der zeitkontinuierlichen PLL sehr gut approximiert wird.

2.4.7 Stabilität

Für die zeitkontinuierliche PLL zweiter Ordnung wurde festgestellt, dass diese im linearen Bereich immer stabil ist solange $\zeta > 0$ gilt. Für zeitdiskrete Systeme stellt die Abtastfrequenz eine neue Größe dar, die sich negativ auf die Stabilität auswirken kann. Ebenso wirken sich große Verzögerungszeiten D negativ auf die Stabilität aus. Wie beim Zeitverhalten zeigt die Stabilitätsanalyse gleiches Verhalten, solange D proportional zum Verhältnis f_s/f_n verändert wird. Die Stabilitätsanalyse wurde mittels numerischer Berechnung der Polstellen durchgeführt. Da sich mit jedem neuen Wert von D eine neue Übertragungsfunktion ergibt, wurde dieser Prozess automatisiert. Ist die Lage der Polstellen bekannt, lautet das notwendige und hinreichende Stabilitätskriterium:

Ein zeitdiskretes System $H(z)$ ist genau dann stabil, wenn sämtliche Pole z_∞ innerhalb des Einheitskreises der z -Ebene liegen ($|z_\infty| < 1$). [Unb00]

Die maximal zulässige Verzögerungszeit, die dieses Stabilitätskriterium erfüllt, wurde empirisch zu

$$D \leq 0,12f_s/f_n \tag{2.84}$$

ermittelt. Solange die wesentlich schärfere Bedingung (2.83) eingehalten wird, ist somit keine Instabilität zu erwarten. In Abbildung 2.9 ist die Stabilitätsgrenze in Abhängigkeit der beiden Parameter Verzögerungszeit D und Eigenfrequenz f_n , normiert zur Abtastfrequenz f_s dargestellt. Ebenso ist der Arbeitsbereich, der sich aus Kriterium (2.83) ergibt, eingezeichnet.

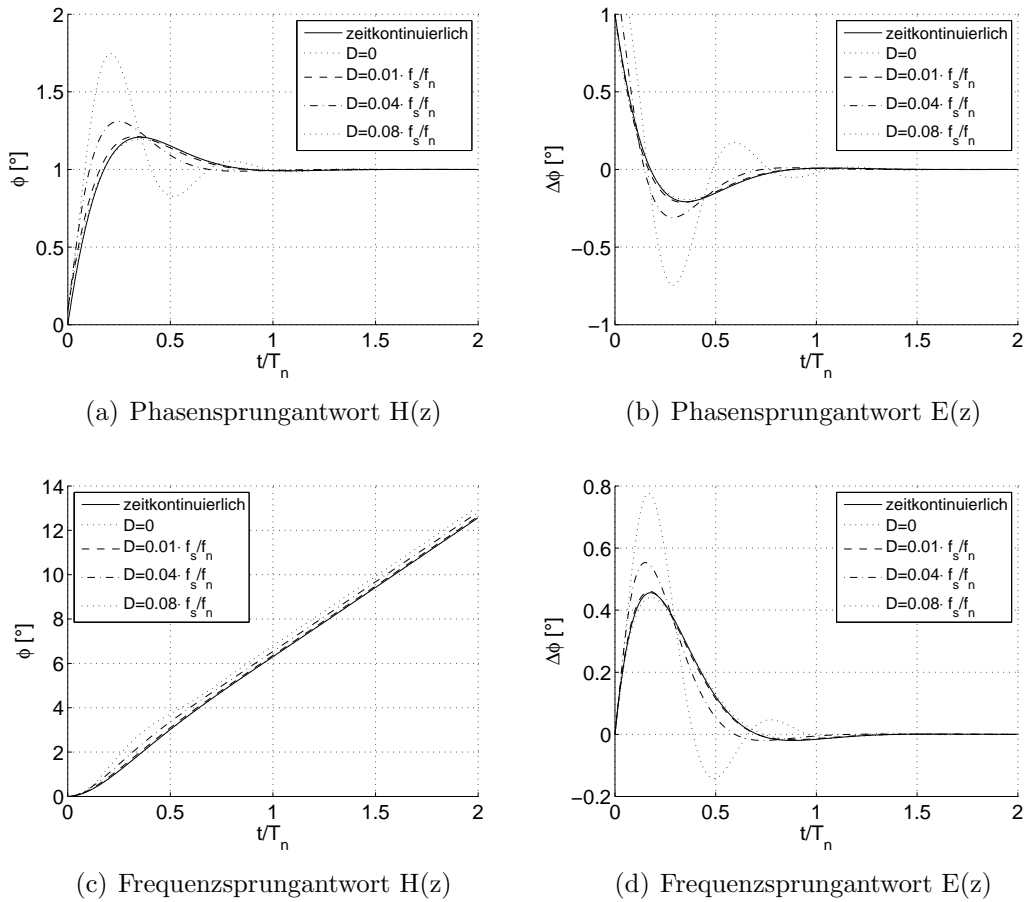


Abbildung 2.8: Phasen- und Frequenz-Sprungantworten von $H(z)$ und $E(z)$

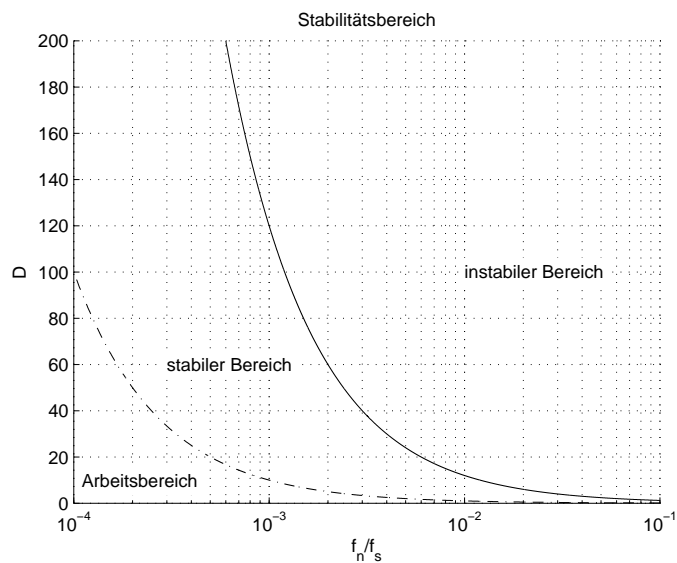


Abbildung 2.9: Stabilitätsdiagramm der zeitdiskreten Typ II PLL mit Ordnung D mit $\zeta=0,707$

2.4.8 Bodediagramm

Das Bodediagramm wurde ebenfalls für verschiedene Verhältnisse von D zu f_s/f_n bei $\zeta = 0,707$ berechnet. Das Ergebnis ist in Abbildung 2.10 dargestellt.

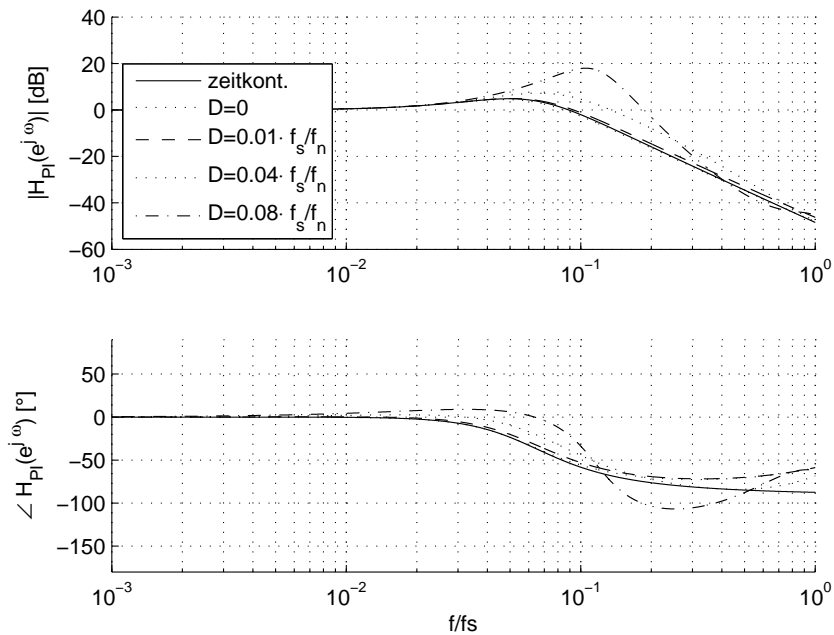


Abbildung 2.10: Bodediagramm der Führungsübertragungsfunktion

2.5 Fang- und Ausrastbereich

In Abschnitt 2.2.2 wurden bereits die unterschiedlichen Arbeits-Frequenzbereiche einer PLL dargestellt. Besonders wichtig ist hierbei der Fangbereich einer PLL, da dieser ein wichtiges Stabilitätskriterium darstellt. Wird die PLL eingeschaltet, während die Frequenz des Eingangssignals außerhalb des Fangbereichs ist, kann nur bedingt gewährleistet werden, ob und in welcher Zeit die PLL einrastet. Prinzipiell gibt zwar die Pull-In-Frequenz Aufschluss über die maximal zulässige Frequenzablage, leider ist diese schwierig oder nur durch Näherungen ermittelbar und stark vom verwendeten Phasendetektor und daher von der verwendeten Architektur abhängig. In der Literatur finden sich Herleitungen für den Fangbereich einer PLL für die unterschiedlichsten PLL-Architekturen. Ausführlich werden PLLs mit Mischer als Phasendetektor [Kro03],[Gar05],[Bes93], sowie solche mit Rechteck-Phasendetektor [Bes93] behandelt. Dabei werden oft Vereinfachungen getroffen, die diese Ergebnisse nicht für andere Ar-

chitekturen nutzen lassen. Daher soll hier ein allgemeiner Ansatz beschrieben werden, der für die lineare PLL zutrifft.

Gesucht ist hierbei die maximale Differenzfrequenz $\Delta\omega_L = \omega_0 - \omega_L$, bei der die PLL sicher einrastet, ohne dass der lineare Arbeitsbereich verlassen wird. D.h. die Differenzphase darf bei einem Frequenzsprung nie die maximal detektierbare Phase des Phasendetektors überschreiten. Formal bedeutet dies

$$\Delta\varphi(t) \leq \varphi_{det,max} \quad \forall t \quad (2.85)$$

bei einem Eingangssignal von

$$\varphi_i(t) = \Delta\omega_L t \quad . \quad (2.86)$$

Der Parameter $\varphi_{det,max}$ ist hierbei die maximal detektierbare Phase des Phasendetektors. Die Vorschrift für die Ermittlung der Fangbereichs-Frequenz lautet somit allgemein

$$\max(\Delta\omega_L t * e(t) + \Delta\varphi(0)) = \varphi_{det,max} \quad , \quad (2.87)$$

wobei mit dem Operator $*$ die Faltungsoperation gemeint ist. Das Signal $e(t)$ ist die Impulsantwort der Fehlerübertragungsfunktion. Der Unterschied zwischen Fang- und Ausrastbereich ist lediglich der, dass beim Ausrastbereich die PLL im eingerastetem Zustand ist ($\Delta\varphi(0) = 0$), während beim Fangbereich eine beliebige Anfangsphase zwischen $-\pi$ und π zulässig ist. Für den linearen Phasendetektor gilt $\varphi_{det,max} = \pi$. Dieser Zusammenhang führt direkt zu der Erkenntnis, dass bei einer Anfangsphase von $\Delta\varphi(0) = \pi$ beim kleinsten positiven Frequenzsprung sofort der lineare Bereich verlassen wird. Die Einfangfrequenz ist somit $\Delta\omega_L = 0$. Daher lassen einige Autoren [Kro03], [SSS05] einen einmaligen Phasenschlupf während des Einfangprozesses zu. Vor diesem Hintergrund liefert die Ausrastfrequenz ein ausreichendes Frequenzintervall, in dem die PLL nach dem Einschalten sicher und mit maximal einem Phasenschlupf einrastet. Im Folgenden soll daher die Ausrastfrequenz für die zeitkontinuierliche PLL mit PI-Regler und linearem Phasendetektor hergeleitet werden.

Für diese gilt

$$\Delta\varphi(t) = \Delta\omega_{PO} t * e_{PI}(t) = \mathcal{L}^{-1}\left\{\Delta\omega_{PO} \frac{1}{s^2} E_{PI}(s)\right\} = \frac{\Delta\omega_{PO}}{N_i \omega_r} e^{-\zeta \omega_n t} \sin \omega_r t \quad (2.88)$$

mit der Resonanzfrequenz $\omega_r = \omega_n \sqrt{1 - \zeta^2}$ [Unb05]. Der Zeitverlauf dieser Funktion wurde bereits in Abbildung 2.5(d) auf Seite 23 dargestellt. Nach Ableiten und Nullsetzen dieser Funktion finden sich die Extrempunkte bei $t = (k\pi + \alpha)/\omega_r$, $k \in \mathbb{Z}_0$ [IB05] mit

$$\alpha = \arctan\left(\frac{\sqrt{1 - \zeta^2}}{\zeta}\right) \quad . \quad (2.89)$$

Für $\zeta > 0$ handelt es sich um eine gedämpfte Schwingung, so dass der erste Extrempunkt dem Maximum der Funktion (bzw. Minimum bei negativem Frequenzsprung) entspricht ($k = 0$). Setzt man diesen Zeitpunkt in (2.88) ein und stellt die Gleichung um, so ergibt sich die Ausrastfrequenz zu

$$\Delta\omega_{PO} = \frac{\varphi_{det,max} \omega_r N_i}{\sin \alpha} e^{\zeta \alpha \omega_n / \omega_r} . \quad (2.90)$$

Man stellt fest, dass die Ausrastfrequenz proportional zur maximal detektierbaren Phase des Phasendetektors, zur Eigenfrequenz der PLL und zum Eingangsfrequenzteiler ist. Außerdem hängt sie von einer Funktion $f(\zeta)$ ab:

$$\Delta\omega_{PO} = \varphi_{det,max} \omega_n N_i f(\zeta) \quad (2.91)$$

mit

$$f(\zeta) = \frac{\sqrt{1 - \zeta^2}}{\sin(\alpha)} e^{\zeta / \sqrt{1 - \zeta^2} \alpha} . \quad (2.92)$$

In Abbildung 2.11 ist diese Funktion dargestellt. Die Funktion hat in diesem Bereich von ζ einen näherungsweise linearen Verlauf und kann durch die Näherungsformel

$$f(\zeta) \approx 1,8(\zeta + 0,5) \quad (2.93)$$

für $0 < \zeta < 2$ abgeschätzt werden. Sie ist ebenfalls in Abbildung 2.11 dargestellt.

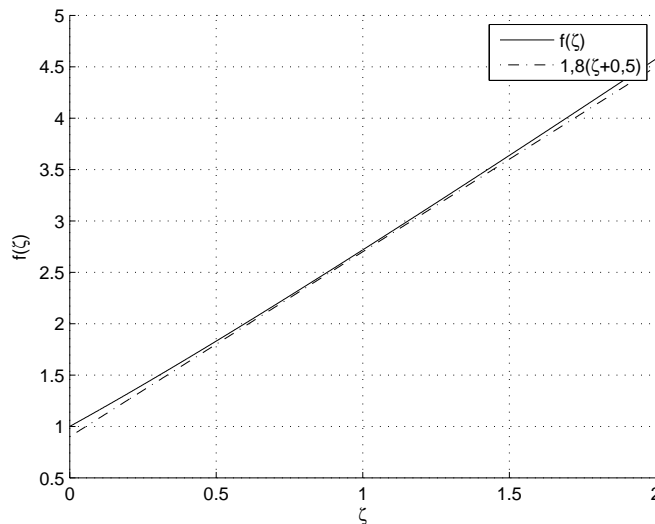


Abbildung 2.11: Abhängigkeit der Ausrastfrequenz von ζ

Der Arbeitsbereich eines gewöhnlichen Phasendetektors beträgt maximal $\varphi_{det,max} = 180^\circ$. Setzt man den in (2.82) ermittelten Wert für die Eigenfrequenz, eine Dämpfung von $\zeta = 0,707$ sowie den Eingangsteiler zu $N_i = 1$ ergibt sich

$$\Delta f_{PO} = 108,6 \text{ kHz} \quad .$$

Dieser Wert liegt weit unterhalb des Frequenzbereichs der PLL. In Abschnitt 2.7.4 wird daher ein Verfahren beschrieben, mit dem der Frequenzbereich erweitert werden kann.

2.6 Einsatz einer PLL als Offset-LO

Um die PLL als Offset-LO nutzen zu können, ist eine Frequenzverschiebung nötig. Abbildung 2.12 zeigt zwei Möglichkeiten, wie diese realisiert werden kann. Der dargestellte „PLL“-Block entspricht der PLL-Struktur nach Abbildung 2.1 auf Seite 12, wobei der Rückkopplungspfad extern ohne Frequenzteilung dargestellt ist.

Die „direkte Methode“ nach Abbildung 2.12(a) entspricht dabei der in Abschnitt 1.5.2 beschriebenen Methode, dass beide Eingangssignale miteinander multipliziert werden und die Summenfrequenz ($f_{RF} + f_{IF}$) durch ein Hochpassfilter herausgefiltert wird. Bei der Mischung tritt das Problem auf, dass Frequenzanteile der Harmonischen der Eingangssignale am Ausgang des Mixers sichtbar werden und nicht mit einem herkömmlichen Filter herausgefiltert werden können. Hier wird die PLL, wie in Abschnitt 2.3.6 beschrieben ist, als schmalbandiges, adaptives Bandpassfilter verwendet.

Bei der „indirekten Methode“ nach Abbildung 2.12(b) arbeitet die PLL bei der variablen Hochfrequenz (f_{RF}). Das Ausgangssignal der PLL wird hierbei um die Zwischenfrequenz (f_{IF}) heruntersgesetzt und die Differenzfrequenz ($f_{RF} + f_{IF} - f_{IF} = f_{RF}$) wird durch ein Tiefpassfilter herausgefiltert. Die PLL rastet somit auf die gewünschte Offset-Frequenz ($f_{RF} + f_{IF}$) ein. Bei beiden Konzepten sollte für ein schnelles Einrasten der PLL die Mittenfrequenz des VCO bzw. DDS auf die Mitte des Frequenzbandes

$$f_c = f_{IF} + (f_{RF,max} - f_{RF,min})/2 \quad (2.94)$$

gesetzt werden.

Die indirekte Methode wird vor allem bei analogen PLLs eingesetzt, da hier keine idealen Mischer existieren und so die Filterung bei der direkten Methode sehr schwierig ist. Außerdem sind die Signalfrequenzen klein, wodurch die Anforderungen an den Frequenzbereich des Phasendetektors geringer sind. Bei zeitdiskreten PLLs kann daher die Abtastfrequenz für diesen Teil prinzipiell geringer gewählt werden. Die direkte Methode hat hingegen bei der zeitdiskreten PLL den Vorteil, dass die Phase der DDS

bereits digital vorliegt und innerhalb der PLL nicht nochmals detektiert werden muss.

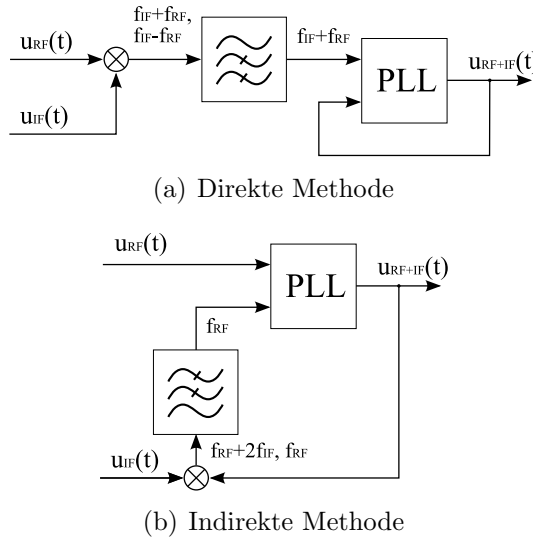


Abbildung 2.12: Methoden zur Offset-Frequenzbildung

Liegen die Signale als analytische Signale vor (siehe kommendes Kapitel 2.7.2), kann statt einem herkömmlichen Mischer mit anschließendem Filter auch ein Einseitenbandmischer verwendet werden, der einer komplexen Multiplikation entspricht.

2.7 Grundlagen der Phasendetektion

2.7.1 Phasendetektoren für rechteckförmige Signale

Gängige Phasendetektoren

Da Rechtecksignale per Definition nur zwei Amplitudenzustände kennen, kann die Bestimmung der Phase nur durch die Auswertung des Zeitpunktes eines Pegelwechsels („0“ nach „1“) erfolgen. Für diese Aufgabe haben sich einige einfache Digitalschaltungen durchgesetzt, die hauptsächlich in zeitkontinuierlichen PLL eingesetzt werden. Die drei am häufigsten verwendeten Schaltungen sind

- Exklusiv-Oder-Gatter für Phasenwerte von -90° bis 90°
- JK-Flipflop für Phasenwerte von -180° bis 180°
- Phasen-Frequenz-Detektor (Abbildung 2.13) für Phasenwerte von -360° bis 360°

Allen drei Digitalschaltungen ist gemein, dass sie ein pulswidenmoduliertes (PWM) Signal erzeugen, dessen Pulsbreite proportional zur Phasendifferenz zweier Eingangssignale ist. Wird dieses PWM-Signal anschließend gefiltert, ergibt sich eine zur Phase proportionale Spannung. Im CMOS PLL IC 74HC4046 sind alle drei Phasendetektoren implementiert, da sie abhängig von den PLL-Parametern unterschiedlich gute Eigenschaften aufweisen. Für eine detaillierte Beschreibung dieser Schaltungen siehe z.B. [Bes93].

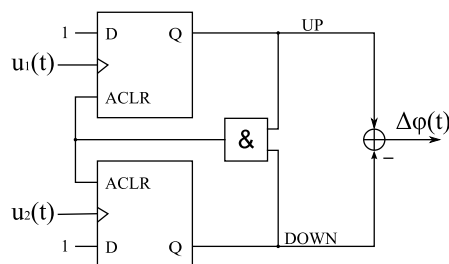


Abbildung 2.13: Aufbau des Phasen-Frequenz-Detektors

Darstellung eines PWM-Signals

Ein PWM Signal trägt seine Information in dem Verhältnis zwischen Puls- und Pausenzeit. Diese Information kann durch Integration über eine Periode zurückgewonnen werden

$$\bar{x}(n) = \frac{1}{T} \int_0^T x(t + nT) dt \quad , \quad (2.95)$$

wobei n die betrachtete Periode des Signals darstellt. Ist das PWM-Signal nicht periodisch oder ist die Periode unbekannt, so kann \bar{x} als zeitkontinuierliches Signal

$$\tilde{\bar{x}}(t) = \frac{1}{\tilde{T}} \int_t^{t+\tilde{T}} x(t) dt \quad (2.96)$$

angenähert werden, solange sich das Signal $\bar{x}(n)$ nur sehr langsam gegenüber der Integrationszeit \tilde{T} ändert und $\tilde{T} \gg T$.

Binärzähler als Phasendetektor mit erweitertem Detektionsbereich

Im Folgenden soll ein Phasendetektor erläutert werden, der auf Binärzählern basiert. In [TS02] ist ein ähnlicher Aufbau mit DACs in einem analogen Phasendetektor als „PLL mit erweitertem Empfangsbereich“ beschrieben. Den Aufbau für zwei Eingangssignale zeigt Abbildung 2.14. Er besteht im Wesentlichen aus zwei Binärzählern

(Addierer und Register), die von den Eingangssignalen getaktet werden. Anschließend wird die Differenz der beiden Zähler gebildet. Alle Zähler arbeiten modulo 2^M , wobei M die Wortbreite bezeichnet. Abbildung 2.15 zeigt beispielhaft die Funktionsweise an zwei Eingangssignalen mit gleicher Frequenz und konstantem Phasenversatz von 90° für eine Wortbreite von $M = 3$. Die Werte der Zähler sind in der Zweierkomplement-Darstellung kodiert. Der Ausgangswert des Phasendetektors ändert seinen Wert maximal um eins und codiert die Phasendifferenz in seiner Pulsbreite von $1/4 \cdot 360^\circ$. Der Ausgangswert der Zähler entspricht einem quantisierten Phasensignal, das aus Frequenzteilung des jeweiligen Eingangssignals um $2^3 = 8$ hervorgehen würde. Dies ist in Abbildung 2.15 gestrichelt eingezeichnet. Die Differenz dieser Signale entspricht exakt dem Mittelwert des Ausgangssignals und ist ebenfalls gestrichelt eingezeichnet.

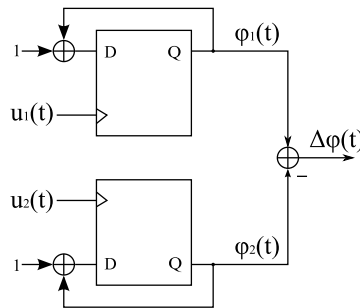


Abbildung 2.14: Aufbau des Binärzähler-Phasendetektors

In Abbildung 2.16 ist das Verhalten bei Eingangssignalen mit unterschiedlicher Frequenz und identischer Nullphasenlage dargestellt. Wieder lässt sich dies anhand der gestrichelt eingezeichneten Phasensignale erklären. Das Ausgangssignal entspricht auch diesmal dem Differenzsignal der Phasensignale, die aus den Eingangssignalen resultieren würden, wenn man sie vorher in der Frequenz durch 8 teilt. Als Phasendetektor einer PLL führt dieses Verhalten zu einem Fehlersignal, welches bis zum Überlauf der Zähler proportional zum Frequenzfehler steigt. Gelingt dem Regler die Korrektur der Frequenz vor einem Überlauf, arbeitet die PLL im linearen Bereich. Die maximal detektierbare Phase beträgt für diesen Phasendetektor

$$\varphi_{max,det} = 2^{M-1} \cdot 2\pi \quad . \quad (2.97)$$

Im Aufbau nach Abbildung 2.14 wurde vereinfachend angenommen, dass die abschließende Subtraktion zeitkontinuierlich erfolgt. Dies ist in digitalen Systemen nicht ohne Übergangswerte möglich. Spätestens für die numerische Integration muss eine Abtastung der Werte erfolgen. Da die Phaseninformation in der Pulsbreite enthalten ist, entsteht dadurch ein unvermeidbarer Fehler. Die Phasenzähler arbeiten völlig

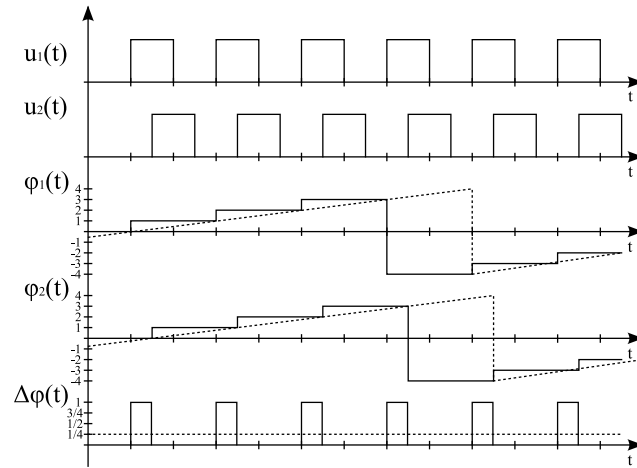


Abbildung 2.15: Zeitverhalten des Binärzähler-Phasendetektors bei Phasenfehler

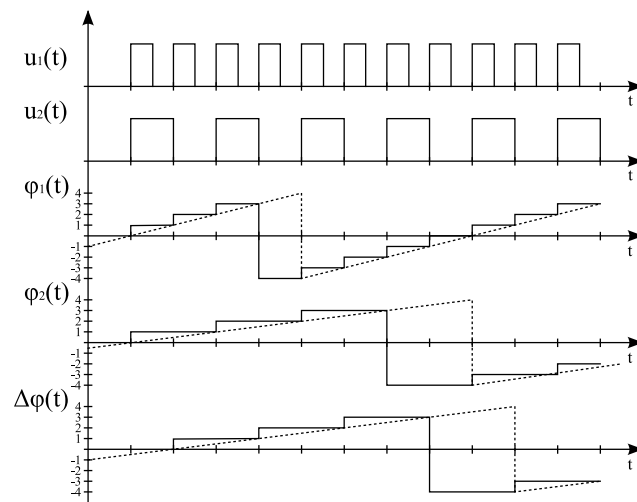


Abbildung 2.16: Zeitverhalten des Binärzähler-Phasendetektors bei Frequenzfehler

asynchron und sind zudem frequenzabhängig. Dies erfordert eine zusätzliche Synchronisationsschaltung. Da es für den Fehler irrelevant ist, ob vor oder hinter den Zählern abgetastet wird, ist eine Abtastung vor den Zählern am günstigsten. Dadurch können die Zähler synchron betrieben werden und müssen genau dann zählen, wenn ein Wechsel von „0“ nach „1“ erkannt wurde. Der Phasenfehler ist abhängig vom Verhältnis zwischen Abtast- und Eingangssignalfrequenz und beträgt maximal

$$\Delta\phi_{max} = 2\pi \frac{f}{f_s} . \quad (2.98)$$

Durch die anschließende Integration kann unter gewissen Umständen der Fehler weiter reduziert werden. Nähere Einzelheiten sind in Kapitel 3.2 zu finden.

2.7.2 Phasendetektoren für sinusförmige Signale

Arcus-Sinus als Phasendetektor

Die Phasendetektion eines sinusförmigen Signals kann als Bestimmung des Arguments einer Sinus-Funktion bei bekannter Amplitude interpretiert werden. Daher scheint es zunächst das Naheliegendste zu sein, die inverse Funktion (Arcus-Sinus) darauf anzuwenden. Diese Methode hat jedoch zwei gravierende Nachteile:

1. Die Arcus-Sinus-Funktion ist mehrdeutig im Bereich $-\pi$ bis π . Es müsste daher anhand der vorherigen Werte entschieden werden, in welchem Quadranten die Phase liegt. Dies kann bei verrauschten Signalen problematisch sein.
2. Die Phasendetektion wäre amplitudenabhängig und nur für eine konstante Amplitude exakt.

Phasendetektion eines analytischen Signals

Ein reelles Singal kann mit Hilfe der Hilbert Transformation in Form eines sogenannten analytischen Signals dargestellt werden [Kam04]. Die Anwendung des analytischen Signals geht zurück auf die analoge Übertragungstechnik mit Einseitenbandmodulation [Bed62] und hat in heutigen digitalen Systemen nicht an Bedeutung verloren. Die Hilbert-Transformation ist im Zeitbereich wie folgt definiert [KK06]:

$$\mathcal{H}\{s(t)\} := \frac{1}{\pi} \int_{-\infty}^{\infty} s(\tau) \frac{1}{t - \tau} d\tau \quad (2.99)$$

Sie kann ebenfalls als Faltung

$$\mathcal{H}\{s(t)\} = s(t) * h_H(t) \quad (2.100)$$

mit der Impulsantwort eines Filters

$$h_H(t) = \begin{cases} 1/(\pi t) & \text{für } t \neq 0 \\ 0 & \text{für } t = 0 \end{cases} \quad (2.101)$$

interpretiert werden. Dieses Filter wird im Folgenden als *Hilberttransformator* bezeichnet. Die Fouriertransformation dieser Impulsantwort liefert

$$H_H(j\omega) = \begin{cases} j & \text{für } \omega < 0 \\ 0 & \text{für } \omega = 0 \\ -j & \text{für } \omega > 0 \end{cases}, \quad (2.102)$$

d.h. der Betrag eines Hilberttransformators ist Eins für alle Frequenzen ungleich Null und die Phase ist konstant -90° für alle positive Frequenzen (bzw. $+90^\circ$ für negative Frequenzen).

Erweitert man ein reelles Signal um einen Imaginärteil, der dessen Hilberttransformierten entspricht, erhält man ein komplexes Signal

$$\underline{s}(t) = s(t) + j\mathcal{H}\{s(t)\} \quad (2.103)$$

Dieses hat die Eigenschaft, dass alle Spektralanteile der Fouriertransformierten für negative Frequenzen verschwinden. Ein solches Signal wird *analytisches Signal* genannt [Kam04].

Das analytische Signal einer sinusförmigen Schwingung der Form

$$s(t) = a(t) \cos(\omega t + \phi_0) \quad (2.104)$$

lautet

$$\underline{s}(t) = a(t) (\cos(\omega t + \phi_0) + j \sin(\omega t + \phi_0)) = a(t)e^{j(\omega t + \phi_0)} \quad (2.105)$$

Die Hüllkurve $a(t)$ und die Phase $\varphi(t) = \omega t + \phi_0$ lassen sich relativ einfach mit Hilfe der komplexen Rechnung ermitteln:

$$a(t) = \sqrt{\operatorname{Re}\{\underline{s}(t)\}^2 + \operatorname{Im}\{\underline{s}(t)\}^2} \quad (2.106)$$

$$\varphi(t) = \arctan \frac{\operatorname{Im}\{\underline{s}(t)\}}{\operatorname{Re}\{\underline{s}(t)\}} (\pm\pi) \quad (2.107)$$

Ein idealer Phasendetektor kann daher als 4 Quadranten-Arcustangens-Abbildung auf das Verhältnis zwischen Imaginär- und Realteil eines aus der Hilberttransformation erzeugten analytischen Signals realisiert werden.

2.7.3 Erzeugung eines analytischen Signals

FIR-Approximation der Hilbert Transformation

Wie im letzten Abschnitt gezeigt wurde, lässt sich die Hilbert-Transformation für zeitkontinuierliche Signale durch eine Faltung mit der Impulsantwort $h_H(t)$ beschreiben. Diese Impulsantwort ist *nichtkausal*, da sie keine zeitliche Begrenzung hat und ist somit nicht realisierbar. Sie kann aber durch eine endliche Impulsantwort approximiert werden. In einem zeitdiskreten System muss außerdem beachtet werden, dass das Spektrum des Hilberttransformators $H_H(j\omega)$ Anteile für $\omega \rightarrow \infty$ enthält und demnach nicht bandbegrenzt ist. Da das zu detektierende Signal in der Regel Bandpassverhalten hat, kann der Hilberttransformator bandbegrenzt werden:

$$H_{H,bb}(j\omega) = \begin{cases} -j & \text{für } 0 < \omega < \omega_g \\ j & \text{für } 0 > \omega > -\omega_g \\ 0 & \text{für ansonsten} \end{cases} \quad (2.108)$$

Unter Berücksichtigung des Nyquist-Theorems lässt sich die Grenzfrequenz zu $\omega_g = \pi f_s$ setzen, was die gesuchte zeitdiskrete Impulsantwort nach Rücktransformation und Abtastung

$$h_{H,bb}(n) = \frac{1 - \cos(\pi n)}{\pi n} = \begin{cases} 2/(\pi n) & \text{für ungerade } n \\ 0 & \text{für gerade } n \end{cases} \quad (2.109)$$

liefert [Kam04]. Die Kausalität lässt sich z.B. mittels Fensterung der Impulsantwort und anschließender Verschiebung in den positiven Zeitbereich erreichen.

Abbildung 2.17 zeigt die verschiedenen Eigenschaften eines approximierten FIR-Hilberttransformators 11. Ordnung, der mit Hilfe der Methode der kleinsten Fehlerquadrate berechnet wurde (`firls()`-Funktion der Matlab[®] Filter Design Toolbox). In Abbildung 2.17(a) ist die Impulsantwort dargestellt. Sie ist ungerade, punktsymmetrisch und jeder gerade Koeffizient ist mit einer Null besetzt. Der Phasengang (Abbildung 2.17(b)) entspricht trotz der Approximation dem Phasengang eines idealen Hilberttransformators. Im Gegensatz dazu zeigt der Betragsfrequenzgang in Abbildung 2.17(c) die für FIR-Filter übliche Welligkeit im Durchlassbereich. Die Interpretation des Hilberttransformators als imaginären Anteil eines komplexen Filters mit Realteil Eins liefert das sogenannte *analytische Filter*

$$\underline{H}(e^{j\omega}) = 1 + jH_{H,bb}(e^{j\omega}) \quad (2.110)$$

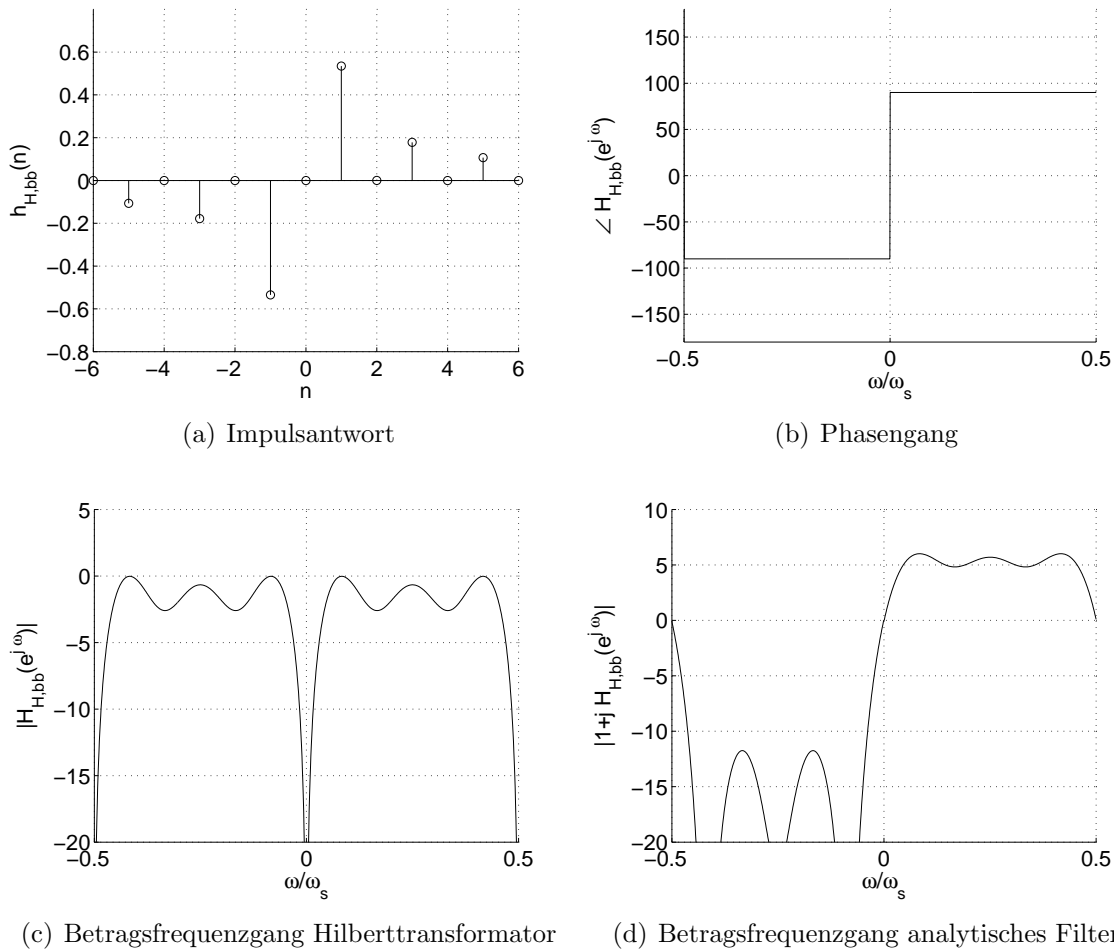


Abbildung 2.17: Eigenschaften eines FIR-Hilberttransformators 11. Ordnung

Abbildung 2.17(d) zeigt den Betragsgang dieses Filters. Im Gegensatz zu einem idealen analytischen Filter ist die Dämpfung der negativen Frequenzkomponenten nicht perfekt. Bei beiden Betragsgängen fällt auf, dass niedrige Frequenzkomponenten schlecht approximiert werden. Anschaulich kann man dies so erklären, dass eine konstante Phasenverschiebung um 90° für eine Frequenz, die gegen Null geht, eine beliebig große Verzögerungszeit bedeutet. Dazu wäre wiederum ein beliebig langes FIR-Filter notwendig. Eine höhere Anzahl an Filterkoeffizienten liefert somit eine höhere Amplituden-Approximationsgüte und einen größeren nutzbaren Frequenzbereich zu niedrigen Frequenzen hin.

Auswirkungen des Amplitudenfehlers der FIR-Hilbert-Approximation auf die Phasendetektion

Während der Realteil eines analytischen Filters perfekt in Amplitude und Phase abgebildet werden kann (konstante Verzögerung), entsteht bei der Bildung des Imaginärteils ein Amplitudenfehler. Die Auswirkungen dieses Fehlers auf den resultierenden Phasenfehler sollen im Folgenden analysiert werden. In Abbildung 2.18(a) ist das analytische Signal einer Sinusschwingung

$$\underline{s} = s_i + js_q = e^{j\varphi} \quad (2.111)$$

in der komplexen Ebene dargestellt. Des Weiteren ist das durch das Hilbert-Filter verfälschte Signal

$$\underline{s}' = s_i + js'_q \quad (2.112)$$

dargestellt. Die Q-Komponente ist durch die frequenzabhängige Verstärkung $G_q \neq 1$ des Hilbert-Filters (siehe Abbildung 2.17(d)) verfälscht worden ($s'_q = G_q s_q$). In der Abbildung ist $G_q = 0.5$. Es ist deutlich zu erkennen, dass sich der Vektor auf einer Ellipse statt auf einer idealen Kreisbahn bewegt. Entsprechend wirkt sich dies auf den Betrag

$$|\underline{s}'(\varphi)| = \sqrt{\cos^2 \varphi + G_q^2 \sin^2 \varphi} \quad (2.113)$$

aus, der in Abbildung 2.18(b) für zwei Umläufe dargestellt ist. Die Auswirkung auf die Phase

$$\angle \underline{s}'(\varphi) = \arctan(G_q \tan \varphi) (\pm \pi) \quad (2.114)$$

ist in Abbildung 2.18(c), zusammen mit dem Phasengang eines idealen Phasendetektors dargestellt. Der Fehler für die Eingangsphase null ist ebenfalls null. Außerhalb der Nullphase wird die Ausgangsphase verzerrt.

In Abbildung 2.18(d) ist der resultierende Phasenfehler für diesen Fall dargestellt. Der Phasenfehler hat einen nahezu sinusförmigen Verlauf, ist symmetrisch um die Nullphase verteilt und die dominante Signalfrequenz entspricht gerade der doppelten Frequenz des Eingangssignals. Da diese in der Regel wesentlich höher liegt als die Eigenfrequenz der PLL ($2\omega_i \gg \omega_n$), lässt sich der Fehler sehr einfach mit einem Tiefpassfilter eliminieren, ohne die Eigenschaften der PLL zu beeinflussen.

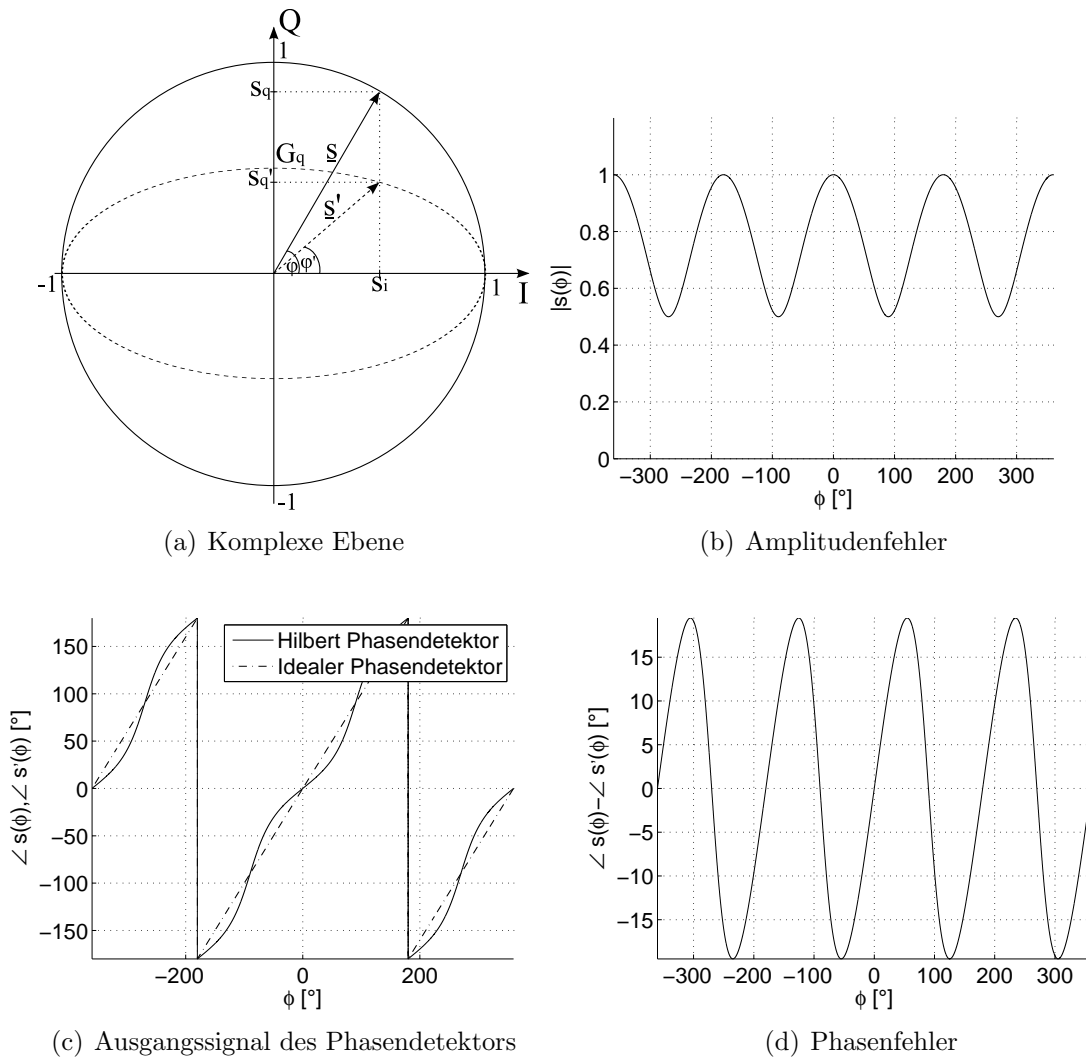


Abbildung 2.18: Auswirkung der Dämpfung des Hilbert-Filters auf den Phasenfehler

2.7.4 Erweiterung des PLL-Einfangbereichs durch Phase Unwrapping

Wie in Abschnitt 2.5 gezeigt wurde, hängt der Einfangbereich einer PLL von Dämpfung, Eigenfrequenz, Eingangsteiler und der maximal detektierbaren Phase des Phasendetektors ab. Die Dämpfung ζ ist oft eine Entwurfsvorgabe und nur in Grenzen variierbar. Ebenso bestimmt die Eigenfrequenz ω_n die Regelbandbreite und somit die Einschwingzeit der PLL. Schnelle Einschwingzeiten sind zwar meistens erwünscht, bei digitalen PLLs führen zu hohe Regelbandbreiten jedoch zu Instabilitäten. Der Wert des Eingangsteilers N_i ist hingegen in großen Bereichen wählbar, da das gewünschte

Ein- zu Ausgangs-Frequenzverhältnis durch Anpassen des Teilers N_o gewählt werden kann. Für PLLs mit Rechteck-Phasendetektoren sind Frequenzteiler sehr einfach durch binäre Teiler implementierbar. Für sinusförmige Eingangssignale ist dies nur durch unverhältnismäßig hohem Aufwand erreichbar (siehe z.B. [Kro03]). Bei digitalen PLLs mit sinusförmigen Eingangssignalen lässt sich unter gewissen Umständen der Bereich der maximal detektierbaren Phase ($\varphi_{det,max}$) relativ einfach durch *Phase-Unwrapping* erweitern.

Das Ausgangssignal eines diskreten 4 Quadranten-Arcustangens Phasendetektors für ein sinusförmiges Eingangssignal mit Kreisfrequenz ω und Anfangsphase φ_0 lautet

$$\varphi(n) = (\omega T_s n + \varphi_0) \bmod 2\pi = (2\pi f / f_s n + \varphi_0) \bmod 2\pi \quad . \quad (2.115)$$

Es entsteht nach jeder Periode ein ungewünschter Phasensprung von 2π nach 0 (bzw. von π nach $-\pi$). Dieser Phasensprung kann detektiert werden, um anschließend die Phase entsprechend zu korrigieren. Um die Höhe eines Sprungs zu ermitteln, wird zunächst die Differenzphase zweier aufeinanderfolgender Abtastwerte ermittelt

$$\Delta\varphi(n) = \varphi(n) - \varphi(n-1) = 2\pi f / f_s \quad . \quad (2.116)$$

Da die maximal darstellbare Signalfrequenz in einem abgetasteten System auf die Nyquistfrequenz $f_{max} = f_s/2$ begrenzt ist, ergibt sich eine maximale Phasenabweichung zwischen zwei Abtastwerten von

$$\Delta\varphi_{max} = 2\pi \frac{f_{max}}{f_s} = \pi \quad . \quad (2.117)$$

Ein Überlauf des Phasendetektors liefert hingegen immer Werte $|\Delta\varphi| > \pi$. Dieses Merkmal kann ausgenutzt werden, um den Zeitpunkt des Phasenüberlaufs zu detektieren und anschließend 2π hinzuzuaddieren. Das gleiche gilt für Phasensignale mit negativer Steigung, die aus einer negativen Differenzfrequenz entstanden sind. Hier kommt es zu einem Phasensprung von 0 auf 2π und es müssen folglich 2π abgezogen werden. Abbildung 2.19 zeigt das Blockschaltbild einer Anordnung, die den Arbeitsbereich eines Phasendetektors nach dieser Methode erweitert.

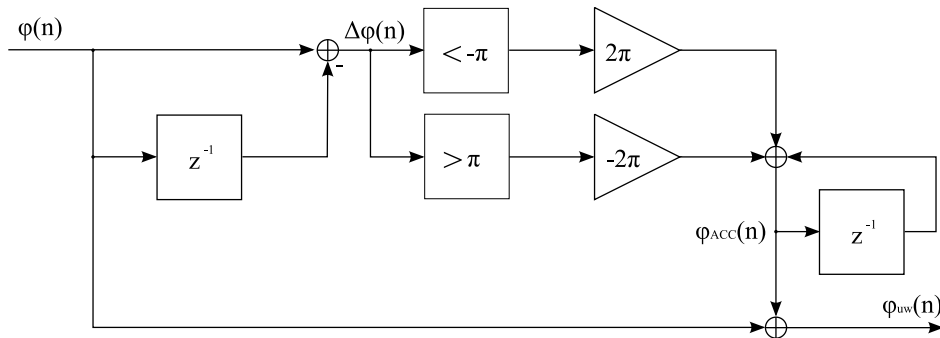


Abbildung 2.19: Aufbau des Phase-Unwrap

Die beiden Vergleiche liefern ein Signal „1“, wenn die Differenzphase größer als π bzw. kleiner als $-\pi$ ist. Dementsprechend wird auf einen Phasenakkumulator (φ_{ACC}) 2π hinzuaddiert oder abgezogen. Der Phasenakkumulator wird permanent der Eingangsphase hinzuaddiert. Tritt kein Phasensprung auf, liefern beide Vergleiche eine „0“, und der Phasenakkumulator hält seinen letzten Wert. In Abschnitt 4.8.5 wird eine effiziente Implementierung dieser Architektur gezeigt.

Mit einer Erweiterung des Phasenwortes um sieben Bit wird die maximal detektierbare Phase um $\varphi_{det,max} = 2^7 \cdot 180^\circ = 23040^\circ$ erweitert. Mit dem in (2.82) ermittelten Wert für die Eigenfrequenz, einer Dämpfung von $\zeta = 0,707$, sowie einem Eingangsteiler zu $N_i = 1$ ergibt sich

$$\Delta f_{PO} = 13,9 \text{ MHz} \quad .$$

Somit arbeitet die PLL immer im linearen Bereich.

3 Realisierungskonzepte

Bei der GSI standen verschiedene Komponenten zur Verfügung, die eine Implementierung des Offset-LO prinzipiell ermöglichten. Im folgenden Abschnitt sollen diese Komponenten, das FPGA Interface Board (FIB) sowie die beiden Erweiterungsplatinen DDS-FIB Adapter Board (DDS-FAB) und ADC/DAC-FIB Adapter Board (ADC/DAC-FAB), in ihren Eigenschaften vorgestellt werden. Anschließend werden in den zwei folgenden Abschnitten unterschiedliche Lösungsansätze vorgestellt, von denen ein Konzept für die Realisierung ausgewählt wurde.

3.1 Beschreibung der zur Verfügung stehenden Komponenten

3.1.1 Das FPGA Interface Board

Das FIB ist eine Platine im Eurokartenformat, die 2005 bei der GSI entwickelt wurde [Kum06b], um die unterschiedlichsten digitalen Schnittstellen zu verarbeiten. Die Platine ist in Abbildung 3.1 dargestellt. Alle Schnittstellen sind mit einem zentralen FPGA der Cyclone[®]-Serie von Altera[®] verbunden (EP1C6Q240C8). Im Folgenden sind die wichtigsten Schnittstellen des FIB mit kurzer Beschreibung aufgelistet

- DSP-Link (paralleles, unidirektionales 8 Bit Interface zum DSP-System)
- μ C-Link (parallele 8 Bit Datenschnittstelle zu einem Mikrocontroller)
- FAB-Interface (parallele 8 Bit Datenschnittstelle für „Huckepack“ Platine (*piggyback*))
- Kontrollsystem-Backplane (parallele Schnittstelle zum Beschleuniger-Kontrollsystem)
- 2 Hochfrequenz-Takteingänge (bis 100 MHz)
- 2 optische HP Versa-Link Sender und Empfänger (seriell, Datenrate bis zu 50 MBit/s)
- RS-232 Schnittstelle zur Parameter-Konfiguration und Diagnose

3 Realisierungskonzepte

Im Folgenden werden die wichtigsten Eigenschaften des verwendeten Altera[®] Cyclone[®] FPGA aufgelistet [Alt07]:

- 5980 Logikzellen (*logic elements*, LE), jedes LE besteht aus einer LUT mit 4 Eingängen, einem Register für D-, T-, JK-, oder SR-Flip-Flops, sowie spezielle Carry-In/Out Leitungen für schnelle Addierer
- *Speed Grade 8* unterstützt max. 275 MHz (interner Takt auf *clock tree*)
- 92160 Bit eingebetteter SRAM (11,520 kByte)
- 185 User-I/O Pins
- 2 PLL-Bausteine
 - statische N_o/N_i -Frequenzteiler mit $N_i = 1$ bis 32 und $N_o = 2$ bis 32
 - Jede PLL hat 2 Ausgänge zum internen *clock tree* und einen Ausgang zu einem dedizierten Pin
 - Jeder Ausgang kann unabhängig in Frequenz und Phasenlage programmiert werden (zur Synthesezeit)
 - Eingangsfrequenz: 15 bis 387 MHz, Ausgangsfrequenz: 15 bis 275 MHz



Abbildung 3.1: FPGA Interface Board

Das FIB kann durch eine zusätzliche „Huckepack-Platine“ erweitert werden (FIB Adapter Board, FAB), zwei dieser Platinen werden in den folgenden Unterkapiteln vorgestellt.

3.1.2 Das DDS-FIB Adapter Board

Das DDS-FIB Adapter Board (DDS-FAB) ist eine Erweiterungskarte, die das FIB um einen DDS-Synthesizer erweitert [MW05]. Abbildung 3.2 zeigt das DDS-FAB, das auf der FIB-Platine aufgesteckt ist. Dieser Aufbau wird u.a. dazu verwendet, das Hochfrequenzsignal für die Beschleunigung zu erzeugen [Elf05], [Kum06a]. Dazu wird der integrierte DDS-Synthesizer Chip AD9854AST von Analog Devices eingesetzt [Ana06]. Die Digital/Analog-Umsetzer sind auf diesem IC bereits integriert. Darüber hinaus sind weitere Einheiten für verschiedene digitale Modulationsarten (FSK, BPSK), ein PLL-Baustein zur Taktfrequenzvervielfachung (Faktor 4 bis 20), ein IQ-Amplituden-Modulator und einige weitere Funktionen integriert.

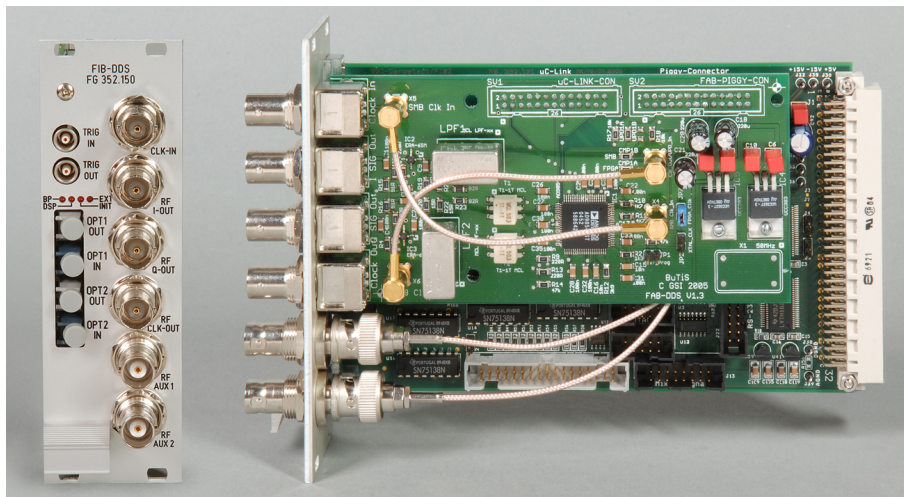


Abbildung 3.2: DDS FIB Adapter Board, aufgesteckt auf die FIB-Platine

Dieser DDS-Baustein kann mit einer Taktfrequenz von bis zu 200 MHz betrieben werden. Die parallele 8 Bit Schnittstelle unterstützt Datenraten von bis zu 95 MByte/s. Das Frequenzwort (FTW) hat eine Wortbreite von 48 Bit. Somit wird bei 200 MHz Taktfrequenz eine Frequenzauflösung von 710nHz erreicht. Die interne Phasenwortbreite der Sinus-/Cosinus-Tabelle beträgt 17 Bit, die Amplitudenwortbreiten der integrierten DACs betragen 12 Bit. Somit werden SFDR-Werte von besser 60 dBc erreicht.

Mit diesen Werten ist diese Leiterplatte prinzipiell für die Anwendung als Offset-LO geeignet. Die Verwendung eines separaten DDS hat jedoch den Nachteil, dass der FPGA keinen direkten Zugriff auf die Phasen- oder Amplitudenwerte hat. Das HF-Ausgangssignal des DDS ist deshalb als phasensynchrones Digitalsignal auf einen FPGA-Pin geführt. Die Eingangssignale des Offset-LO werden über die beiden Hochfrequenz-Takteingänge mit Hilfe von Komparatoren phasentreu in Digital-

signale umgesetzt. Somit sind dem FPGA alle Phaseninformationen in Form von Rechtecksignalen zugänglich.

3.1.3 Das ADC/DAC-FIB Adapter Board

Das ADC/DAC-FAB dient ebenfalls zur Erweiterung des FIB, welches dies um zwei Analog/Digital- und zwei Digital/Analog-Umsetzer ergänzt [San06]. Die Platine ist in Abbildung 3.3 dargestellt. Die A/D-Wandler (ADC) vom Typ LTC2254 haben eine Auflösung von 14 Bit und erlauben Abtastraten von bis zu 105 MSPS. Die D/A Wandler (DAC) AD9744 haben ebenfalls eine Auflösung von 14 Bit und können mit bis zu 210 MSPS verwendet werden. Es erreicht SFDR-Werte von typisch 55 dBc im DAC-Kanal und ein Signal-Rausch-Verhältnis (SNR) von typisch 55 dB im ADC-Kanal. Mit diesen Werten ist die Leiterplatte ebenfalls für die direkte Verarbeitung der Offset-LO Signale geeignet. Allerdings muss dafür die DDS-Komponente im FPGA realisiert werden. Der Vorteil daran ist, dass die Phaseninformation im FPGA vorhanden ist und nicht erst detektiert werden muss. Die Eingangskanäle liefern quantisierte sinusförmige Signale, was eine präzise Phasendetektion ermöglicht.

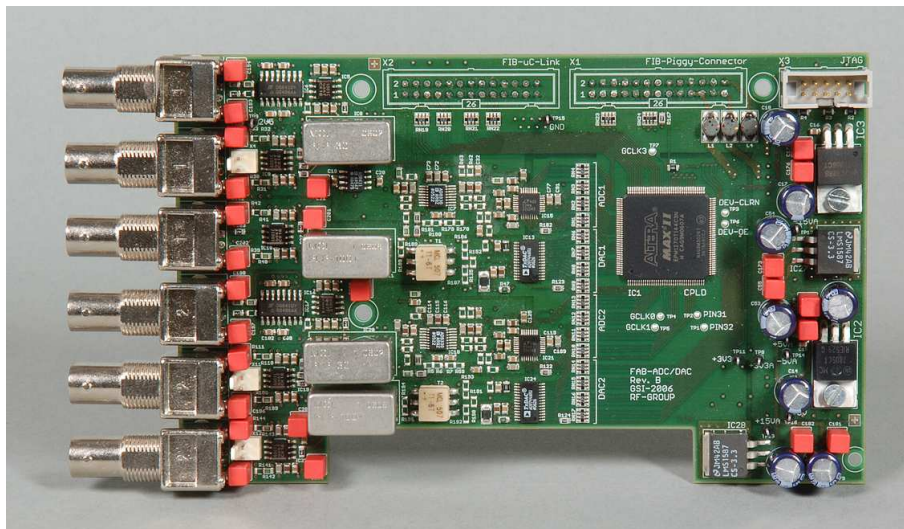


Abbildung 3.3: ADC/DAC FIB Adapter Board

Zur Verbindung der ADC und DAC über einen 16 Bit Datenbus wird der MaxII[®] CPLD EPM1270T144 von Altera[®] verwendet. In diesem sind das Busprotokoll sowie einige Register zur Steuerung weiterer Funktionen implementiert. Dieses CPLD ist von der Architektur dem Cyclone[®] FPGA sehr ähnlich. Er verfügt über 1270 Logikelemente, die vom Aufbau identisch zu den Cyclone[®] Logikelementen sind. Mit dem

verwendeten *speed-grade* 3 sind Taktfrequenzen bis zu 304 MHz möglich [Alt06]. Allerdings ist das begrenzende Element bei dieser Anwendung nicht die Abtastrate der Wandler, sondern die Übertragungsrate der Bus-Schnittstelle. Bei voller Datenrate zweier ADC und eines DAC müssten $2 \cdot 14 \text{ Bit} \cdot 105 \text{ MHz} + 14 \text{ Bit} \cdot 210 \text{ MHz} = 5810 \text{ MBit/s}$ mit niedrigen Verzögerungszeiten übertragen werden. Bei der bisherigen Implementierung waren Datenraten von 1920 MBit/s ($16 \text{ Bit} \cdot 120 \text{ MHz}$) in einer Richtung möglich, ein Umschalten der Treiber in die andere Richtung benötigte jedoch mehrere Takte, was die Datenrate für die PLL-Implementierung auf 640 MBit/s reduzierte. Wie in Abschnitt 4.2 noch gezeigt wird, konnte die Datenrate bei gleichzeitigem Lesen und Schreiben durch Umstrukturierung des Busses auf 1920 MBit/s erweitert werden. Wie später ebenfalls noch gezeigt wird, kann die Datenrate durch Unterabtastung des schmalbandigen Zwischenfrequenz-Signals weiter reduziert werden, so dass diese Bandbreite für die Anwendung ausreichend ist.

3.2 Realisierungskonzept eines Offset-LOs mit FIB und DDS-FAB

Wie im vorletzten Abschnitt erwähnt wurde, stehen dem FIB unter Verwendung des DDS-FAB alle Signale in Form von rechteckförmigen Signalen zur Verfügung. In Kapitel 2.7.1 wurden bereits Phasendetektoren für rechteckförmige Signale vorgestellt. Der dabei vorgestellte Phasendetektor mit Binärzählern ist prinzipiell auch für mehrere Eingangssignale verwendbar. Abbildung 3.4 zeigt den Aufbau einer solchen Offset-PLL. Die Blöcke „CNT PD“ enthalten nur die Phasenakkumulatoren. Die Subtraktion ist extern eingezeichnet. Damit die PLL auf die Summenfrequenz einrastet, ist es notwendig, das Summensignal der Eingangs-Phasensignale von dem Ausgangs-Phasensignal abzuziehen. Der Aufbau entspricht im Wesentlichen dem grundlegenden Aufbau einer PLL aus Abbildung 2.1 auf Seite 12. Die Phasendifferenz $\Delta\varphi$ gelangt auf den PI-Regler $F_{PI}(z)$. Dieser Regler erfüllt in dieser Anordnung zwei Aufgaben: Er ist Schleifenfilter der PLL und Integrator des PWM-Signals der Phasendetektoren. Mit dem Frequenzwort FTW_0 kann die Mittenfrequenz der PLL eingestellt werden. Das daraus berechnete FTW wird über ein paralleles Bus-Interface (Block „Bus Arb.“) an das DDS-IC AD9854 gesendet, welches das LO-Signal erzeugt. Da das Businterface nicht über die nötige Bandbreite verfügt um die Phase der DDS zurück zu übertragen, muss die Phase des LO-Signals detektiert werden.

In Kapitel 2.7.1 wurde bereits der resultierende Phasenfehler in Abhängigkeit von Signalfrequenz und Abtastfrequenz für den abgetasteten Binärzähler-Phasendetektor angegeben. Bei einer Signalfrequenz von maximal 26,8 MHz und einer abgeschätzten, maximal möglich Abtastfrequenz von 200 MHz resultiert ein Phasenfehler von $\pm 24,12^\circ$. Dieser Wert ist zunächst inakzeptabel.

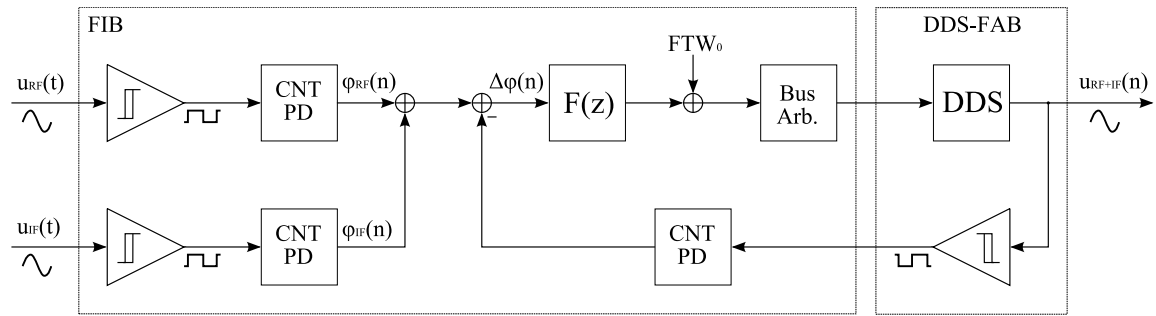


Abbildung 3.4: Aufbau mit Binärzähler-Phasendetektor

Durch die anschließende Integration lässt sich dieser Fehler jedoch weiter reduzieren. Dazu ist es allerdings erforderlich, dass der Fehler gleichmäßig um den tatsächlichen Phasenwert verteilt ist. Simulationen dieser PLL-Struktur haben gezeigt, dass dies normalerweise der Fall ist und dass dadurch der Fehler auf unter ein Grad reduziert werden kann. Aufgrund der einfachen Struktur wurde eine Test-Implementierung ohne Offset-Addition vorgenommen. Diese Implementierung hat gezeigt, dass eine PLL mit dieser Architektur möglich ist und funktioniert. Leider fiel bei den anschließenden Messungen auf, dass ein Spezialfall im Konzept nicht berücksichtigt wurde. Die Fehlerreduktion durch Integration funktioniert solange, wie der Phasenfehler in etwa gleichverteilt um die tatsächliche Phasenlage ist. Entspricht jedoch die Abtastfrequenz einem ganzzahligen Vielfachen der Eingangssignalfrequenz, wird das Eingangssignal immer an der gleichen Stelle abgetastet. Dies führt zu einem konstanten Phasenfehler, der durch die Integration nicht weiter reduziert werden kann. Sobald ein kleiner Frequenzversatz existiert, resultiert daraus eine Schwebung zwischen Abtastzeitpunkt und Zeitpunkt des Nulldurchgangs des Eingangssignals. Solange die Periode der Schwebungsfrequenz kleiner als die Integrationszeit ist, kann sichergestellt werden, dass der Phasenfehler durch Integration deutlich reduziert werden kann. Da der Offset-LO eine Frequenzdynamik von mehreren Oktaven haben soll, findet sich keine Abtastfrequenz, die diese Eigenschaft erfüllt. Damit schied diese einfache Art der Implementierung aus Mangel an Phasengenauigkeit für den Offset-LO aus.

3.3 Realisierungskonzept eines Offset-LOs mit FIB und ADC/DAC-FAB

Im Gegensatz zur DDS-FAB-Platine, bei der alle Signale in Form von Rechtecksignalen dem FPGA zugeführt werden, ist die ADC/DAC-FAB-Platine in der Lage, quantisierte Amplitudenwerte der Signale zu liefern. Wie später noch gezeigt wird, konnte über das Businterface eine Abtastfrequenz der beiden ADC-Kanäle von je-

weils 30 MHz erreicht werden. Da nur einer der beiden DAC verwendet wird, beträgt die Abtastfrequenz für diesen 60 MHz. Bis auf das Zwischenfrequenzsignal reicht die Abtastrate aus, die Signale direkt im Basisband darstellen zu können. Da die ZF eine konstante Frequenz hat und somit praktisch keine Bandbreite belegt, ist für dieses Signal eine Unterabtastung zulässig. Für die Verarbeitung innerhalb der PLL muss dieses jedoch in der Abtastfrequenz wieder hochgesetzt werden. Dies ist durch Interpolation, im Sinne der Multiratenverarbeitung, mit anschließender Filterung möglich [CR83], [Fli93].

Die maximale Signalfrequenz beträgt 26,8 MHz, daher beträgt die minimale Abtastfrequenz innerhalb der PLL 53,6 MHz. Eine Abtastratenerhöhung um ganze Vielfache der ursprünglichen Abtastfrequenz ist einfach zu realisieren. Daher ist eine Erhöhung der ADC-Abtastfrequenz um mindestens den Faktor 2 erforderlich. Prinzipiell ist eine möglichst hohe Abtastrate wünschenswert, da sich Stabilität und Genauigkeit der PLL verbessern. Sie kann aber nicht beliebig erhöht werden, da die Taktfrequenz des FPGA limitiert ist. Als Kompromiss wurde eine Abtastfrequenz von 120 MHz gewählt, da diese auf dem verwendeten FPGA relativ einfach erreicht werden kann und dem Vierfachen der ADC-Abtastfrequenz entspricht.

Abbildung 3.5 zeigt den kompletten Aufbau einer solchen Realisierung. Der FPGA des FIB ist mit dem CPLD des ADC/DAC-FAB über einen Bus verbunden. Die beiden Bus-Arbiter-Blöcke (Bus-Arb.) sorgen dafür, dass die Daten der Wandler transparent zum FPGA des FIB übertragen werden. Anschließend erfolgt die zuvor erwähnte Abtastratenerhöhung mit den Blöcken RF-/IF-Upsampling. Aus diesen werden analytische Signale erzeugt, um anschließend die Frequenzverschiebung nach der „direkten Methode“ aus Kapitel 2.6 mit einem komplexen Multiplizierer (Einseitenbandmischer) zu realisieren. Das frequenzverschobene, analytische Signal gelangt nun auf einen Block ADPLL-IQ, der in Abbildung 3.6 dargestellt ist.

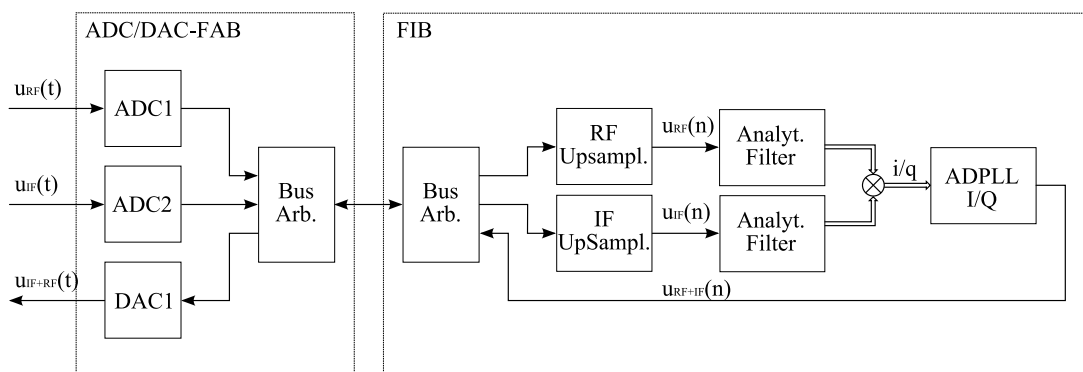


Abbildung 3.5: Aufbau der All-digital PLL

Der Aufbau dieses Blocks entspricht im Wesentlichen dem Aufbau aus Abbildung 2.1. Frequenzteiler sind in dieser nicht erforderlich, und statt eines VCO kommt eine DDS zum Einsatz. Die Phasendetektion ist nur für das komplexe Eingangssignal nötig. Außerdem enthält dieser Aufbau schon eine optionale Phase-Unwrap-Komponente (PU), sowie ein weiteres Tiefpassfilter ($F_{LP}(z)$) zur Unterdrückung der in Abschnitt 2.7.3 beschriebenen, hochfrequenten Phasenfehler. Das Ausgangssignal der PLL wird auf umgekehrtem Weg über die Bus-Arbitrierung transparent an den DAC geleitet.

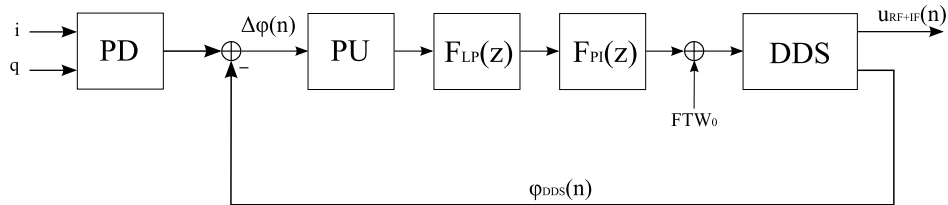


Abbildung 3.6: Aufbau der ADPLL-IQ Komponente

4 Implementierung

4.1 Entwurfsablauf

Schon während der ersten Detaillierung der Konzeption war es hilfreich, ein abstraktes Simulationsmodell zu entwickeln, das aus idealen Komponenten besteht. Dafür wurde Matlab[©] (The MathWorks) in Verbindung mit der Simulink[©]-Erweiterung verwendet. Des Weiteren kamen die folgenden Werkzeuge zum Einsatz:

- Control System Toolbox - Zur Berechnung von Frequenzgängen und Impulsantworten
- Signal Processing Toolbox - Zur Berechnung der Signalspektren
- Filter Design Toolbox - Zur Berechnung und Optimierung von zeitdiskreten Filtern
- Fixed-Point Toolbox - Zur Quantisierung von zeitdiskreten Filtern
- Filter Design HDL Coder - Zur VHDL-Code-Generierung von zeitdiskreten Filtern

Als Erweiterung von Simulink[©] wurde das *Signal Processing Blockset* zur Einbindung von Filtern und Up-/Downsampling-Blöcken verwendet.

Trotz der umfangreichen Erweiterungen haben einige Standard-Elemente zur Simulation von PLLs gefehlt. Daher wurde eine eigene PLL-Bibliothek für Simulink[©] entwickelt, die in Kapitel 5.1 näher beschrieben wird. Nach der Simulation mit idealen Komponenten erfolgte die Implementierung der Einzelkomponenten in VHDL. Die Simulation wurde auf Bitebene mit dem Standard-VHDL-Simulator Modelsim (Mentor Graphics) durchgeführt. Nach erfolgreicher Implementierung wurden die VHDL-Komponenten mit Hilfe des *Altera DSP Builder Blockset* in ein Simulink-Modell übersetzt. Mit diesem konnte die Simulation der gesamten PLL mit der tatsächlichen Implementierung wiederholt werden (inklusive Wortbreiteneffekte, zusätzlichen Pipeline-Verzögerungen, etc.). Zwischenstände wurden mit der Altera[©] Software Quartus[©] II synthetisiert und in der Schaltung getestet. Während der Entwicklung erfolgten Synthesedurchläufe, um den Ressourcenaufwand sowie die maximale Taktfrequenz abschätzen zu können. Da der verwendete FPGA über keine dedizierten

Multiplizierer verfügt, musste stets darauf geachtet werden, dass die verwendeten Komponenten möglichst ohne Multiplizierer auskommen.

4.2 Interface zu ADC/DAC-FAB

Die Busschnittstelle zwischen FIB und ADC/DAC-FAB (vgl. Abbildung 3.5) besteht aus einem 16 Bit Datenbus, einem 8 Bit Adressbus sowie einem 6 Bit Steuerbus. Der Datenbus ist über Bustreiber (SN74LVTH16245ADL) mit dem CPLD verbunden. In der ursprünglichen Implementierung des CPLD war das Businterface als bidirektionaler Bus ausgeführt. Für den Einsatz als PLL müssen ständig ADC-Werte gelesen und DAC-Werte geschrieben werden, was eine häufige Umschaltung der Datenrichtung notwendig macht. Damit in keinem Takt zwei Treiberausgänge kurzgeschlossen werden, ist es notwendig, die Treiber der Reihe nach umzuschalten. Bei einer Richtungsänderung vom Lesebetrieb in den Schreibbetrieb bedeutet dies zuerst den CPLD-Treiber in Leserichtung zu schalten, dann den externen Datentreiber umzuschalten und im folgenden Takt den FPGA Treiber als Ausgang umzuschalten. Für eine Abfolge ADC1 lesen, ADC2 lesen, DAC1 schreiben, etc. wie es für die PLL nötig ist, sind mit diesem Verfahren 9 Takte notwendig, wovon alleine 6 Takte für die Richtungsumschaltung benötigt werden. Zwar wäre eine Übertragung mit FIFO-Zwischenspeichern deutlich effizienter, sie würde aber zusätzliche Verzögerungszeiten bedeuten. Bei einer Taktfrequenz von 120 MHz wäre so die Abtastfrequenz auf 13,33 MHz limitiert. Für eine erforderliche Bandbreite von 4,6 MHz (5,4 MHz-0,8 MHz) reicht diese nach dem Nyquist-Theorem zwar gerade noch aus, allerdings wäre eine Abtastratenumsetzung auf FIB und FAB notwendig.

Daher wurde der Bus auf zwei 8 Bit Busse aufgeteilt. Diese lassen sich völlig unabhängig in ihrer Richtung betreiben und gestatten beide vollen Zugriff auf alle Register des FAB. Der Adress- und Steuerbus wurde aufgeteilt auf jeweils 4 und 3 Bit. Für die PLL-Anwendung kann ein Bus in Schreib- und der andere Bus in Leserichtung geschaltet werden. Zwar werden dann für einen Zugriff auf ein 14 Bit Datenwort der Wandler zwei Takte benötigt, dafür sind keine Takte für die Richtungsumschaltung erforderlich. Bei 120 MHz Bustakt beträgt die ADC-Rate 30 MSPS für zwei Kanäle und die DAC-Rate für einen Kanal 60 MSPS. Somit reicht die Abtastrate des DAC aus, um direkt den Frequenzbereich von 22,2 bis 26,8 MHz darzustellen. Der Frequenzbereich von 0,8 bis 5,4 MHz des HF-Signals kann ebenfalls direkt dargestellt werden. Nur die Zwischenfrequenz von 21,4 MHz muss somit unterabgetastet und auf dem FIB wieder heraufgesetzt werden. Damit die PLL-Implementierung bei einer konstanten Abtastrate betrieben werden kann, wird ebenfalls das HF-Signal in der Abtastrate hochgesetzt. Für andere Anwendungen erlaubt das neue Busprotokoll ebenso wie das alte Protokoll die volle Datenrate zu einem Wandler, indem beide 8 Bit Busse in die gleiche Richtung geschaltet werden.

4.3 Zahlendarstellung

Die Darstellung der Zahlenwerte erfolgt durchgehend im Zweierkomplement. Da in den einzelnen Berechnungsschritten immer wieder die Wortbreite angepasst werden musste, wurde die folgende Darstellung für gebrochene Zahlen verwendet:

$$A_B = a_{N-1}, a_{N-2}a_{N-3}\dots a_1a_0 \quad (4.1)$$

Der Wert der Dezimalzahl ergibt sich somit aus

$$A_D = -a_{N-1} + \sum_{i=0}^{N-2} a_i 2^{i-N+1} . \quad (4.2)$$

Die Position des Kommas direkt nach dem MSB erlaubt zum einen Wertebereich von

$$A_D \in [-1\dots 1 - 2^{-(N-1)}] , \quad (4.3)$$

zum anderen kann die Wortbreite durch Abschneiden der niederwertigen Bits verringert werden, ohne den Zahlenwert selbst zu verändern (abgesehen von der Darstellungsgenauigkeit der resultierenden Auflösung). Dies ist oft bei Ergebnissen der Addition und Multiplikation nötig.

In der VHDL-Implementierung wurde stets darauf geachtet, dass eine interne Wortbreite als VHDL-generic vorgegeben werden konnte, um zur Synthesezeit den Kompromiss aus Platzbedarf, Geschwindigkeit und Genauigkeit beeinflussen zu können.

4.4 Aufbau der ADPLL

Die beiden Abbildungen 4.1 und 4.2 geben die gesamte Implementierung der ADPLL wieder. Die Implementierung der einzelnen Komponenten erfolgt im Detail in den kommenden Unterkapiteln.

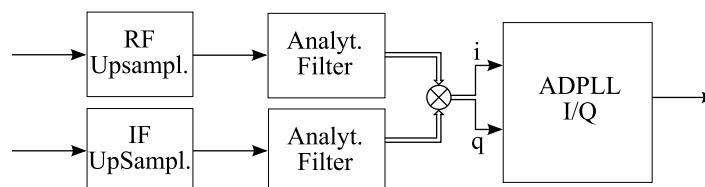


Abbildung 4.1: Gesamtaufbau der ADPLL

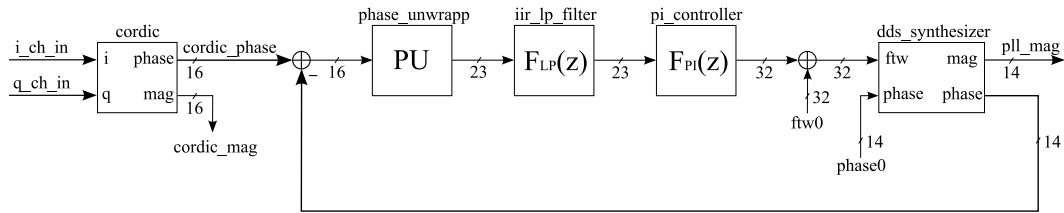


Abbildung 4.2: Aufbau der Entity adpll_iq

4.5 Implementierung der Interpolationsfilter

4.5.1 Anforderungen an die Interpolationsfilter

Das Prinzip einer Abtastratenerhöhung erfolgt in zwei Stufen, die in Abbildung 4.3 dargestellt sind. In der ersten Stufe wird das Signal durch einen „Interpolator“ in der Abtastrate heraufgesetzt und anschließend in der zweiten Stufe gefiltert. Bei einer Erhöhung der Abtastrate um den Faktor R ($R \in \mathbb{N}$) werden $R-1$ Nullen zwischen jedem Abtastwert eingefügt. Das nachfolgende Filter dient zur Interpolation der erforderlichen Zwischenwerte.

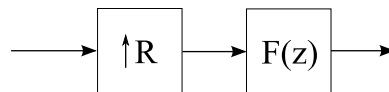


Abbildung 4.3: Aufbau der Abtastratenerhöhung

Die Interpolation kann als Multiplikation mit einem Impulskamm der Periode R interpretiert werden. Dieser „Mischprozess“ sorgt für eine Spiegelung des Originalspektrums um die Frequenzlinien $r/R \cdot f_s$ ($r \in \{1..R-1\}$), wobei f_s die bereits heraufgesetzte Abtastrate bezeichnet ($f_s = Rf'_s$). Dies soll im Folgenden anhand der Spektren der Offset-LO-Eingangssignale veranschaulicht werden. Das Spektrum des um den Faktor $R = 4$ heraufgesetzten HF-Signals ist in Abbildung 4.4(a) dargestellt. Das ursprüngliche Spektrum reicht bis zur Abtastfrequenz $f'_s = 30$ MHz, wurde um die Nyquist-Frequenz (15 MHz) gespiegelt und hat daher Signalanteile von 0,8 bis 5,4 MHz, sowie von 24,6 MHz bis 29,2 MHz ($30 \text{ MHz} - (0,8..5,4) \text{ MHz}$). Bei dem um den Faktor vier heraufgesetzten Signal spiegelt sich das originale Spektrum bei 30 MHz, das resultierende Spektrum bei 60 MHz und das daraus resultierende Spektrum bei 90 MHz. Zur eindeutigen Interpolation ist daher ein Tiefpassfilter notwendig, das nur für das Originalspektrum, sowie für den um die neue Nyquist-Frequenz gespiegelten Frequenzbereich von 114,6 bis 119,2 MHz, durchlässig ist. Der Frequenzgang des Filters ist in Abbildung 4.4(a) gestrichelt eingezeichnet.

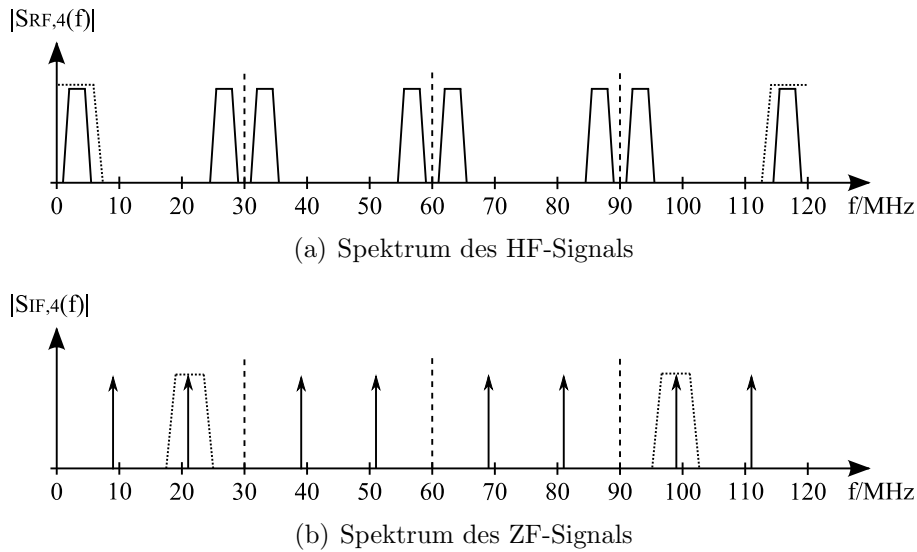


Abbildung 4.4: Spektren der um den Faktor 4 heraufgesetzten Eingangssignale

Ähnlich sieht dies für das ZF-Signal aus, dessen Spektrum in Abbildung 4.4(b) dargestellt ist. Im Gegensatz zum HF-Signal wurde dieses unterabgetastet, daher soll nicht die untere Signalfrequenz von 9,6 MHz interpoliert werden, sondern die um die ursprüngliche Nyquist-Frequenz von 15 MHz gespiegelte Komponente von 21,4 MHz. Daher muss für die Interpolation des unterabgetasteten Signals ein Bandpassfilter verwendet werden, welches idealerweise den gestrichelt eingezeichneten Verlauf haben sollte.

Die Abtastratenerhöhung kann, solange R keine Primzahl ist, in mehreren Stufen erfolgen ($R = R_1 \cdot R_2 \cdot \dots$). Dies ist in der Implementierung vorteilhaft, da schmalbandige Filter bei einer niedrigen Abtastfrequenz einfacher zu implementieren sind, als gleichwertige Filter bei einer hohen Abtastfrequenz.

4.5.2 Frequency Sampling Filter

Damit die Phaseninformation erhalten bleibt, ist es notwendig, dass die Interpolationsfilter linearphasig sind. Da konventionelle FIR-Filter sehr aufwendig werden können, wurden für diesen Zweck spezielle *frequency sampling filter* (FSF) verwendet. Ein klassisches FSF besteht aus einem Kammfilter (engl. *comb*) sowie einem oder mehreren IIR-Filtern [Tay83]. Das Kammfilter erzeugt Nullstellen, die gleichmäßig auf dem Einheitskreis verteilt sind. Die zusätzlichen IIR-Filter werden so gewählt, dass sie Polstellen erzeugen, die ebenfalls auf dem Einheitskreis liegen und Nullstellen kompensieren. Gelingt dies exakt, kürzen sich Pol- und Nullstelle in der Übertragungsfunktion heraus, und es resultiert ein FIR-Filter. Gelingt die Kompensation

nicht exakt, kann es vorkommen, dass eine Polstelle außerhalb des Einheitskreises liegt und das System instabil wird. Erfolgt die Implementierung im Zweierkomplement, kann eine exakte Kompensation erreicht werden [MB95]. In [MB04] wurden einige IIR-Filter beschrieben, die ohne Multiplizierer auskommen, was die resultierenden FSF extrem attraktiv für die Interpolationsaufgabe macht.

Ein zeitdiskretes Kammfilter mit der Übertragungsfunktion

$$F_{zD}(z) = 1 - z^{-Dz} \quad (4.4)$$

erzeugt D_z gleichmäßig auf dem Einheitskreis verteilte Nullstellen bei

$$z_{0d} = e^{jd/D_z 2\pi} \quad . \quad (4.5)$$

In Tabelle 4.1 sind Koeffizienten sowie die Ordnung D_p für die IIR-Übertragungsfunktion

$$C_p(z) = \frac{z^{-D_p}}{a_0 + a_1 z^{-1} + \dots + a_{D_p} z^{-D_p}} \quad (4.6)$$

angegeben, die bei den angegebenen Winkeln ϕ_1 und ϕ_2 Polstellen auf dem Einheitskreis haben. Durch Auswahl eines oder mehrerer IIR-Filter können so die Durchlassfrequenzen ($f = \phi/360^\circ \cdot f_s$) definiert werden. Das Kammfilter muss anschließend nur so dimensioniert werden, dass es Nullstellen an den entsprechenden Positionen hat. Je mehr Nullstellen das Kammfilter hat, desto schmalbandiger ist das resultierende Filter. Durch Verwendung mehrerer, hintereinander geschalteter FSF kann die gewünschte Sperrdämpfung erzielt werden.

$C_k(z)$	D_p	a_0	a_1	a_2	a_3	a_4	ϕ_1	ϕ_2
$C_1(z)$	1	1	-1				0°	
$C_2(z)$	1	1	1				180°	
$C_3(z)$	2	1	1	1			$\pm 120^\circ$	
$C_4(z)$	2	1	0	1			$\pm 90^\circ$	
$C_6(z)$	2	1	-1	1			$\pm 60^\circ$	
$C_5(z)$	4	1	1	1	1	1	$\pm 72^\circ$	$\pm 144^\circ$
$C_8(z)$	4	1	0	0	0	1	$\pm 45^\circ$	$\pm 135^\circ$
$C_{10}(z)$	4	1	-1	1	-1	1	$\pm 36^\circ$	$\pm 108^\circ$
$C_{12}(z)$	4	1	0	-1	0	1	$\pm 30^\circ$	$\pm 150^\circ$

Tabelle 4.1: Koeffizienten der Polfilter mit Winkel der Polstelle bis zur Ordnung vier [MB04]

Die Funktionsweise soll nun an einem Beispiel erläutert werden. Wählen wir aus der Tabelle das Filter $C_3(z)$ mit den Polstellen bei $\pm 120^\circ$ aus. Ein Kammfilter mit $D = 6$

hat Nullstellen bei $0^\circ, \pm 60^\circ, \pm 120^\circ$ und 180° . Abbildung 4.5(a) zeigt den Betragsfrequenzgang des Kammfilters. Die Dämpfung wird für Frequenzen, die Vielfache von $f_s/6$ sind, unendlich. Abbildung 4.5(b) zeigt die Übertragungsfunktion des IIR-Filters welche bei zwei diskreten Frequenzen $\pm 1/3 f_s$ unendlich hohe Verstärkung aufweist.

Die Reihenschaltung der beiden Filter ergibt sich nach Multiplikation und Polynomdivision der beiden Übertragungsfunktionen zu

$$F_{z6c3}(z) = (1 - z^{-3}) \cdot \frac{z^{-2}}{1 + z^{-1} + z^{-2}} = z^{-2} - z^{-3} \quad . \quad (4.7)$$

Man erkennt, dass die Gesamtübertragungsfunktion trotz rekursiver Strukturen eine endliche Impulsantwort hat. Der entsprechende Betragsfrequenzgang ist in Abbildung 4.5(c) dargestellt. Das zugehörige Pol-/Nullstellen-Diagramm vor der Kürzung, ist in Abbildung 4.5(d) zu sehen.

Eine weitere Vereinfachung kann durch Ausnutzung der sogenannten *noble identity* für Interpolatoren verwendet werden. Die Operation der Interpolation ist i.A. nicht kommutativ, die Vertauschung der Reihenfolge einer Übertragungsfunktion $F(z)$ mit einem Interpolator ($\uparrow R$) liefert [CR83]

$$(\uparrow R)F(z) = F(z^{1/R})(\uparrow R) \quad . \quad (4.8)$$

D.h. wenn die Filterung vor der Interpolation vorgenommen wird, resultiert ein um den Faktor R kürzeres Filter. Dies ist natürlich nur bei Filtern sinnvoll, bei denen die Substitution $F(z^{1/R})$ zu einer Übertragungsfunktion mit ganzzahligen Potenzen von z resultiert. Für das Kammfilter $F_{zD}(z)$ ist dies für den Fall $D_z/R \in \mathbb{Z}$ möglich und resultiert in

$$F_{zD}(z^{1/R}) = 1 - z^{-\frac{D_z}{R}} \quad . \quad (4.9)$$

Somit werden R Register eingespart. Die resultierende Struktur zeigt Abbildung 4.6.

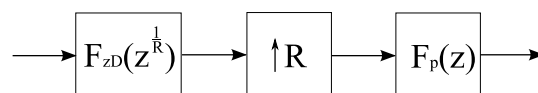
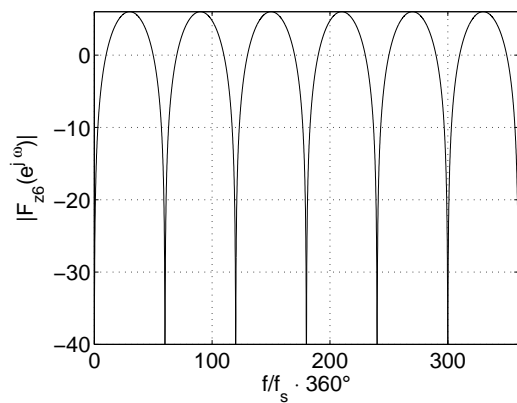
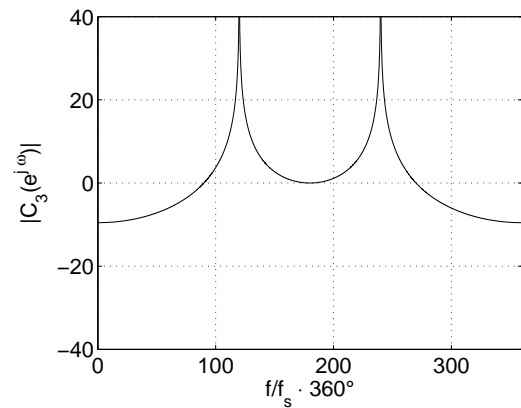


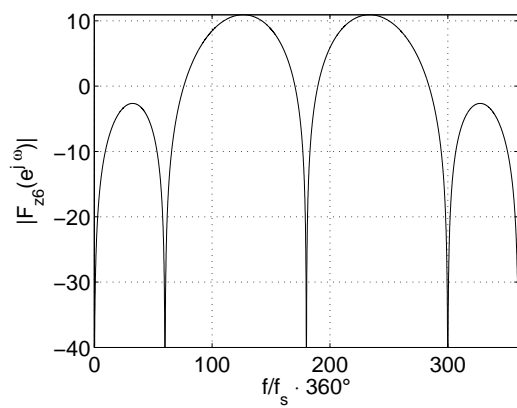
Abbildung 4.6: Abtastratenerhöhung mit FSF-Filtern



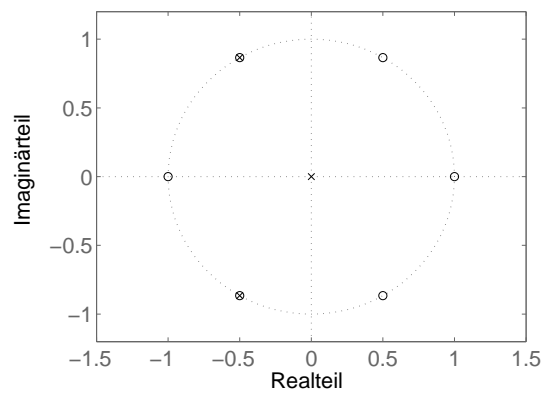
(a) Betragsfrequenzgang $F_z(e^{j\omega})$ für $D=6$



(b) Betragsfrequenzgang $F_p(e^{j\omega})$



(c) Betragsfrequenzgang $F_p(e^{j\omega}) \cdot F_z(e^{j\omega})$

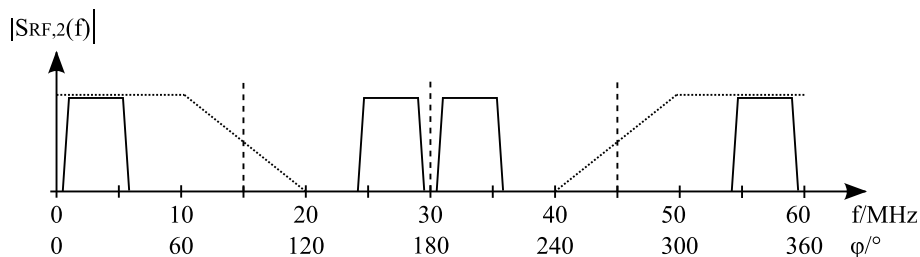


(d) Pol-/Nullstellen-Diagramm $F_p(e^{j\omega}) \cdot F_z(e^{j\omega})$

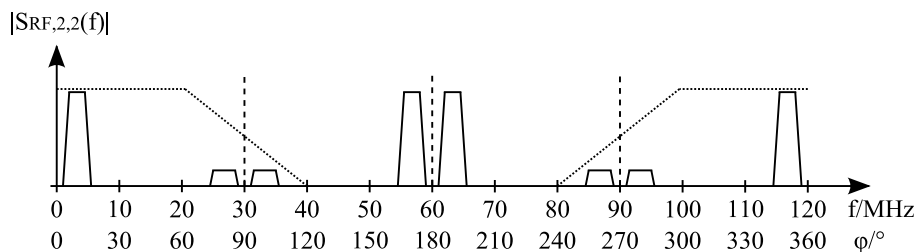
Abbildung 4.5: Betragsfrequenzgänge und Pol-/Nullstellen-Diagramm $F_{z6c3}(z)$

4.5.3 Entwurf der Interpolationsfilter

Die Interpolation von $f'_s = 30$ MHz auf $f_s = 120$ MHz erfolgt in zwei Stufen. Abbildung 4.7(a) zeigt das Spektrum nach der ersten Stufe der Abtastratenerhöhung von 30 MHz auf 60 MHz. Ein Tiefpassfilter ist notwendig, um das Spektrum des Originalsignals zu interpolieren. Unter den Frequenzangaben sind die jeweiligen Winkel des Einheitskreises angegeben. Das benötigte Filter muss ab ca. 20 MHz sperren, daher muss es bei 0° und $\pm 60^\circ$ durchlässig sein. Das bereits behandelte FSF-Kammfilter mit $D = 6$ hat Nullstellen bei $0^\circ, \pm 60^\circ, \pm 120^\circ$ und 180° . Mit den IIR-Filtern $C_1(z)$ und $C_6(z)$ aus Tabelle 4.1 können die Nullstellen bei 0° und $\pm 60^\circ$ eliminiert werden. Um die benötigte Dämpfung zu erlangen, wurden drei dieser Filter in Serie verwendet. Da das resultierende Filter eine Verstärkung von 216 (46,69 dB) aufweist, wurde das Eingangswort um 8 Bit verschoben, um eine Division durch $2^8 = 256$ zu erzielen, damit die Verstärkung unter Eins bleibt. Die Wortbreite des Filters ist dafür entsprechend zu erhöhen. Der Frequenzgang des resultierenden Filters ist in Abbildung 4.10(a) dargestellt. Die Spiegelfrequenzen werden um mehr als 70dB gedämpft. Der Dämpfungsunterschied im Nutz-Frequenzbereich beträgt etwa 4,5 dB. Die Anforderungen an das Filter nach der zweiten Abtastratenerhöhung sind in Abbildung 4.7(b) zu sehen. Dadurch, dass die Spiegelfrequenzen ausreichend unterdrückt wurden, kann das Filter der ersten Stufe ebenfalls hier verwendet werden. Es hat bei 24,6 MHz bereits 25dB Dämpfung.



(a) Spektrum nach erster Abtastratenerhöhung



(b) Spektrum nach zweiter Abtastratenerhöhung

Abbildung 4.7: Spektren der zweistufigen Interpolation des HF-Signals

Analog erfolgt die Entwicklung der ZF-Filter. In Abbildung 4.8 sind die Anforderungen an beide Filter der beiden Stufen angegeben. Beide Filter benötigen jeweils einen Durchlassbereich bei 21,4 MHz. In beiden Filtern kann wieder das FSF-Kammfilter mit $D = 6$ verwendet werden. In der ersten Stufe muss das IIR-Filter einen Pol bei 120° aufweisen, was mit $C_3(z)$ aus Tabelle 4.1 realisiert wird. Bei der zweiten Stufe wird ein Pol bei 60° benötigt. Dies ist mit $C_6(z)$ möglich. Um das Sperrverhalten zu verbessern, wurden ebenfalls je drei Filter in Serie betrieben. Da die beiden Filter eine Verstärkung von jeweils 43,63 (32,8 dB) aufweisen, wurde das Eingangswort bei diesen um 6 Bit verschoben. Die Frequenzgänge der beiden Filter sind in den Abbildungen 4.10(b) und 4.10(c) dargestellt. Die implementierte Struktur ist in Abbildung 4.9 wiedergegeben.

4.6 Implementierung des analytischen Filters

Die Grundlagen des analytischen Filters wurden bereits in Abschnitt 2.7.3 beschrieben. Die Implementierung besteht aus zwei Unterkomponenten, dem Hilbert-Filter zur Erzeugung des Imaginärteils sowie einer Verzögerungs-Komponente, die den Realteil um die Laufzeit des Hilbert-Filters verzögert. Die Laufzeit des Hilbert-Filters beträgt $(N - 1)/2$ Takte. Bei einer ungeraden Koeffizientenanzahl ergibt sich somit eine ganzzahlige Anzahl an Verzögerungstakten, die einfach durch Register kompensiert werden kann. Zusätzlich können damit Verzögerungen, die durch Pipeline-Register im Hilbert-Filter entstehen, kompensiert werden.

Das Hilbert-Filter wurde mit der Methode der kleinsten Fehlerquadrate, der Matlab[®] Filter Design Toolbox (`fir1s()`-Funktion), berechnet. Diese Funktion kann mit dem Parameter 'hilbert' direkt ein Hilbert-Filter erzeugen. Zur Berechnung des Filters müssen Vektoren mit Frequenz- und Amplitudenstützpunkten übergeben werden. Mit einem weiteren Vektor kann eine Gewichtung der Frequenzbereiche erfolgen. Als Frequenzbereichsvorgabe wurde ein symmetrischer Frequenzbereich von $0,005f_s$ bis $0,995f_s$ gewählt ($f_s = 120$ MHz). Die Symmetrie führt zu einer Impulsantwort, bei der alle geraden Koeffizienten mit einer Null besetzt sind. Mit einer Amplitudenvorgabe von 0,84 wurde dafür gesorgt, dass die maximale Verstärkung auch an Stellen maximaler Welligkeit immer kleiner als 0 dB ist. Die Filterlänge wurde fest auf $N = 11$ gesetzt, da ein Filter dieser Größe auf dem verwendeten FPGA neben den anderen Komponenten noch realisierbar ist.

Multiplikationen müssen natürlich nur für Multiplikatanden ungleich Null durchgeführt werden. Davon gibt es für $N=11$ genau 6. Durch die Punktsymmetrie ($h(n) = -h(-n)$) können nochmals die Hälfte aller Multiplikationen eingespart werden, indem die Ergebnisse mehrfach verwendet werden. Die Architektur des so optimierten Filters ist in Abbildung 4.11 zu sehen.

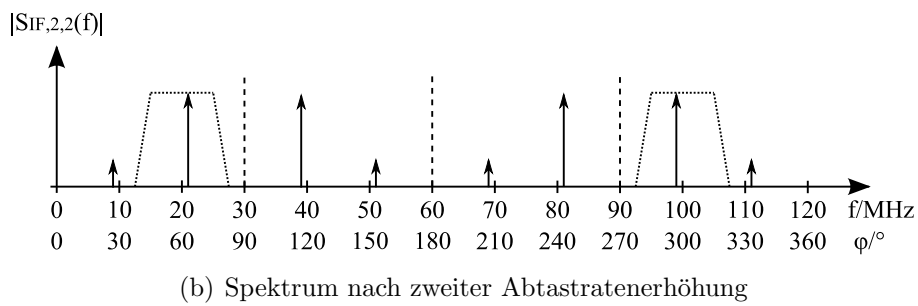
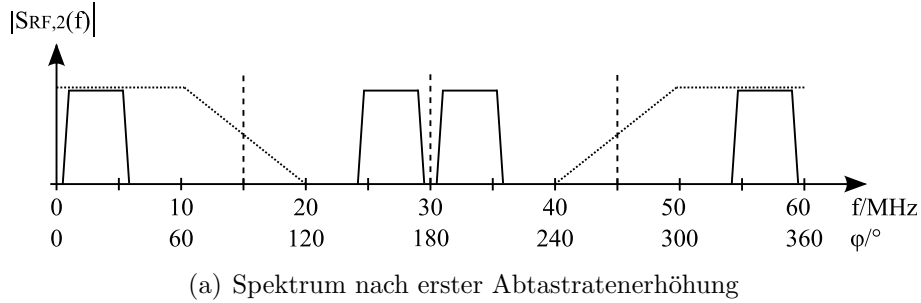


Abbildung 4.8: Spektren der zweistufigen Interpolation des ZF-Signals

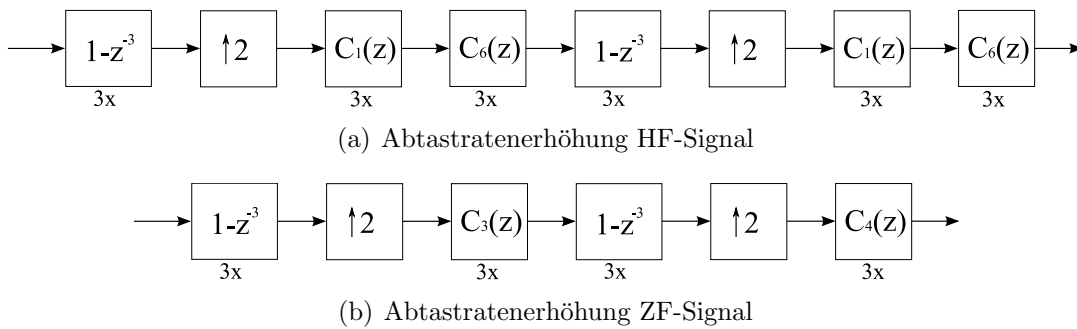
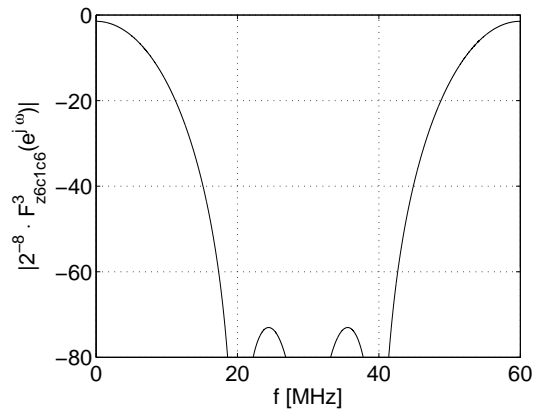
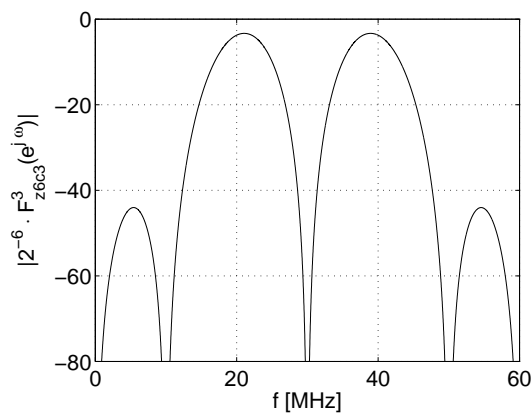


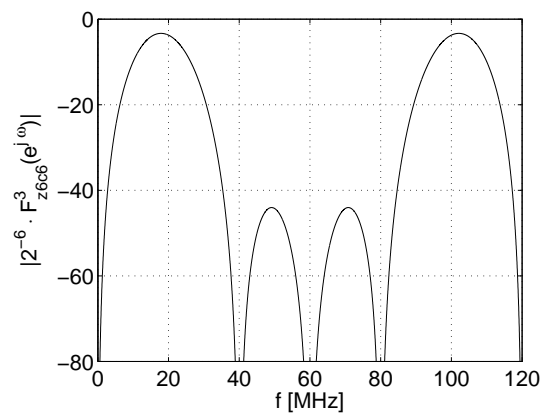
Abbildung 4.9: Struktur der implementierten Abtastratenerhöhung



(a) $1/2^8 \cdot |F_{z6c1c6}^3(e^{j\omega})|$



(b) $1/2^6 \cdot |F_{z6c3}^3(e^{j\omega})|$



(c) $1/2^6 \cdot |F_{z6c6}^3(e^{j\omega})|$

Abbildung 4.10: Betragsfrequenzgänge der HF- und ZF-Filter

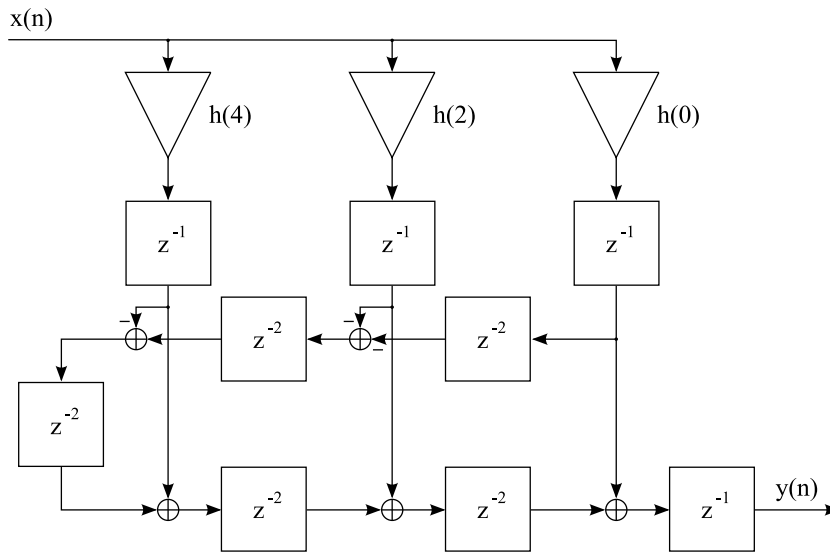


Abbildung 4.11: Aufbau des Hilbert-Filters

Mit Hilfe der Matlab[©] „Filter Design HDL Coder“ Toolbox lässt sich aus berechneten Filtern der Filter Design Toolbox VHDL- oder Verilog-Code generieren. Für Festkomma-Implementierungen müssen die Koeffizienten zuvor mit Hilfe der Fixed-Point Toolbox in eine Festkomma-Darstellung konvertiert werden. Die „Filter Design HDL Coder“ ist in der Lage, punktsymmetrische Filter zu implementieren (Filter-Typ `dfasymfir`). Leider ist hierbei die Platzierung der Pipeline-Register so ungünstig, dass die erforderliche Abtastfrequenz von 120 MHz nicht realisierbar war. Nach Konvertierung in ein *direct form* FIR Filter (Filter-Typ `dffir`) war die Platzierung der Pipeline-Register günstiger. Es benötigte durch die vielen Multiplizierer jedoch 1121 LE (18,75% des verwendeten FPGA) und erreichte nur 98 MHz, obwohl 5 Pipeline-Stufen verwendet wurden. Nachdem im letzten Schritt der Implementierung das Design nicht mehr auf dem FPGA platziert werden konnte, wurde dieses Filter von Hand neu implementiert und dabei nach Abbildung 4.11 optimiert. Das Ergebnis benötigte nur noch 336 LE (5,6%) bei zwei Takten Verzögerung und maximal 248 MHz Taktfrequenz.

Das optimierte Hilbert-Filter kann in Eingangs-, Ausgangs- und interner Wortbreite frei konfiguriert werden. Um das Ergebnis zu überprüfen, wurde die VHDL-Komponente mit dem DSP Builder[©] in ein Simulink-Modell übersetzt und mit dem idealen Filter verglichen (Abbildung 4.12). Um den Frequenzgang zu ermitteln, wurde ein Einheitspuls (Dirac-Impuls) auf das Simulink-Modell gegeben und anschließend die Antwort mit einer FFT in Matlab[©] untersucht. Abbildung 4.13 zeigt das Ergebnis der so ermittelten Frequenzgänge des implementierten analytischen Filters mit einer internen Wortbreite von 16 Bit und 32 Bit. Die Ein- und Ausgangswortbreiten betragen jeweils 16 Bit. Außerdem ist der Frequenzgang des idealen analytischen Filters

dargestellt. Die niedrige interne Auflösung von 16 Bit resultiert in einer um lediglich 0,6 dB schlechteren Dämpfung für negative Frequenzkomponenten gegenüber der 32 Bit Variante. Daher wurde eine interne Darstellung von 16 Bit gewählt.

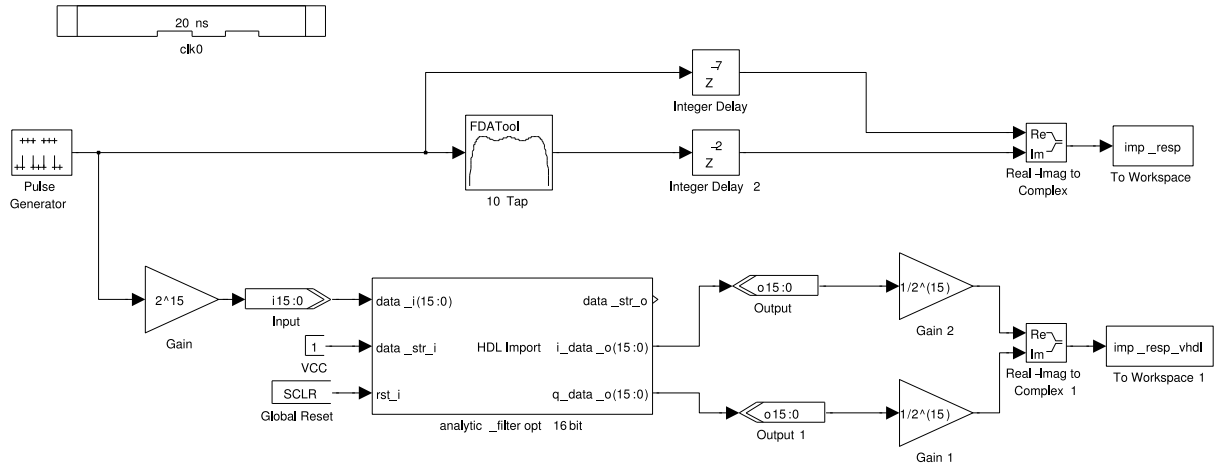


Abbildung 4.12: Simulationsaufbau des analytischen Filters

4.7 Implementierung der komplexen Multiplikation

Die grundlegende mathematische Operation eines Einseitenbandmischers ist die komplexe Multiplikation zweier analytischer Signale. Die einfache komplexe Multiplikation liefert hierbei das obere Seitenband, die Multiplikation mit dem konjugiert komplexen liefert als Ergebnis das untere Seitenband. Da die Ausgabefrequenz der Offset-PLL der Summe aus HF- und ZF-Frequenz entsprechen soll, wird das obere Seitenband benötigt. Die komplexe Multiplikation zweier Signale liefert

$$(i_1 + jq_1) \cdot (i_2 + jq_2) = i_1i_2 - q_1q_2 + j(i_1q_2 + i_2q_1) \quad (4.10)$$

Diese Operation benötigt vier reelle Multiplikationen sowie zwei reelle Additionen. Durch elementare Umformungen lässt sich diese zu

$$(i_1 + jq_1) \cdot (i_2 + jq_2) = i_2(i_1 + q_1) - \underbrace{q_1(i_2 + q_2)}_{=u} + j(q_2(1_1 - q_1) + \underbrace{q_1(q_2 + i_2)}_{=u}) \quad (4.11)$$

umstellen [Win05], die durch Ausnutzung des doppeltes Vorkommens der Variablen u nur noch drei reelle Multiplikationen und fünf reelle Additionen benötigt. Nach jeder dieser Operationen wurde in der Implementierung ein Pipelineregister hinzugefügt. Da sich die meisten Operationen parallel ausführen lassen, beträgt die Verzögerung durch die Pipeline-Register drei Takte.

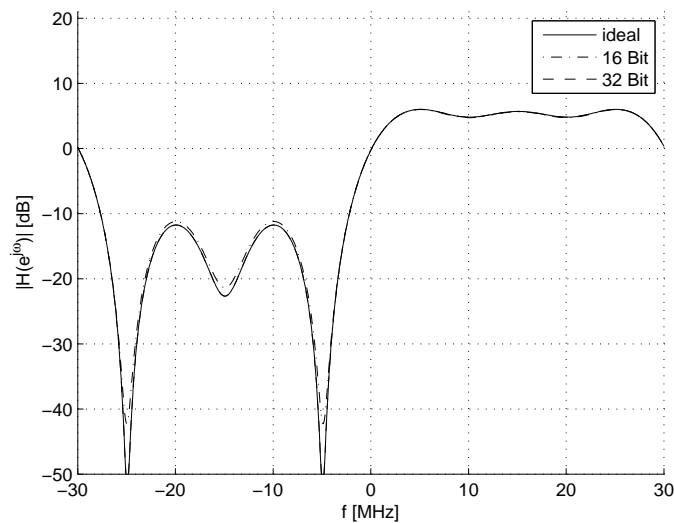


Abbildung 4.13: Betragsfrequenzgang des implementierten analytischen Filters

4.8 Implementierung der PLL-Komponenten

4.8.1 Phasendetektor

Die Phasendetektion kann, sobald ein analytisches Signal vorliegt, auf die Berechnung eines 4-Quadranten-Arcustangens aus dem Verhältnis zwischen Imaginär- und Realteil zurückgeführt werden (2.107). Eine elegante Möglichkeit für diese Berechnung liefert der CORDIC-Algorithmus. Dieser wurde bereits 1959 von J. E. Volder vorgestellt [Vol59] und später von J. S. Walther weiterentwickelt [Wal71]. Der Algorithmus kann in drei Modi verwendet werden, zirkular, linear und hyperbolisch. Im zirkularen Modus lassen sich Betrag und Phase eines kartesischen Vektors berechnen, wobei keine Multiplikationen verwendet werden. Der im Grunde sequentielle Algorithmus kann durch Aufrollen der Schleife parallelisiert werden. Durch den Einsatz von Pipelining-Techniken kann ein hoher Durchsatz erzielt werden. Im Rahmen der Kooperation der GSI mit dem Fachgebiet MES der TU Darmstadt wurde dieser Algorithmus für den FPGA des DSP-Systems implementiert [GZMGK06]. Diese VHDL-Komponente erwartet zwei Eingangsvektoren mit einer Wortbreite von jeweils 16 Bit und berechnet die Amplitude und Phase mit einer Genauigkeit von $3 \cdot 10^{-3}$ bzw. $0,05^\circ$. Somit ist diese für die PLL ideal geeignet und konnte fast unmodifiziert übernommen werden. Die einzige Modifikation war die Entfernung einer Vorverarbeitungsstufe, die nur für das DSP-System verwendet wurde. Obwohl sie ursprünglich für einen Xilinx-FPGA entwickelt wurde, ließ sie sich ohne Probleme mit dem Altera-Synthesewerkzeug übersetzen. Die Implementierung verwendet 16 Pipeline-Stufen für den CORDIC-Algorithmus sowie zwei Takte für die Vor- und Nachverarbeitung.

4.8.2 Direct Digital Synthesizer

Die Grundlagen der direkten digitalen Synthese wurden bereits in Abschnitt 2.4.2 erläutert. Die Implementierung des Synthesizers ist in Abbildung 4.14 dargestellt und repräsentiert die Realisierung des Aufbaus von Abbildung 2.6 auf Seite 25. Die wesentlichen Unterschiede sollen im Folgenden erläutert werden. Das Register des Phasenakkumulators wurde verschoben, um es als Pipelineregister zu verwenden. Ebenso ist ein weiteres Pipelineregister für die Addition des Phasenoffsets hinzugekommen. Die Wortbreiten der Frequenz-, Phasen- und Amplitudenauflösung sind per VHDL-generic frei konfigurierbar und sind in der Abbildung entsprechend abgekürzt. N entspricht der Frequenzauflösung (generic `ftw_width`), M der Phasenauflösung (generic `phase_width`) und P der Amplitudenauflösung (generic `amplitude_width`).

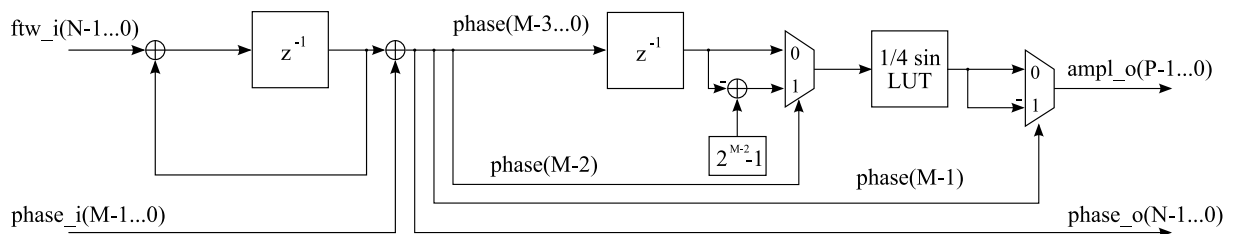


Abbildung 4.14: Blockschaltbild der implementierten DDS

Ausnutzung der Symmetrie der Sinusfunktion

Es wurde nur die Sinus-Tabelle implementiert, da für die PLL nur ein Kanal erforderlich ist. Für die Speicherung der Sinus-LUT lassen sich die Symmetrien der Sinus-Funktion ausnutzen. Negative Argumente (Quadranten III und IV) können durch die Punktsymmetrie

$$\sin(-x) = -\sin(x) \quad (4.12)$$

aus Werten der positiven Quadranten I und II ermittelt werden. Weiterhin gilt die Symmetrie

$$\sin(\pi/2 + x) = \sin(\pi/2 - x) \quad (4.13)$$

D.h. Werte des zweiten Quadranten können durch Anpassung des Argumentes aus Werten des ersten Quadranten ermittelt werden. Somit reicht es aus, lediglich den ersten Quadranten in der LUT zu speichern.

Das Phasenwort `phase` wird im Zweierkomplement nach Formel (4.1) entsprechend mit π multipliziert dargestellt. Somit ergibt sich ein Wertebereich der Phase von

$$\varphi = phase \cdot \pi \in \left[-\pi \dots \pi - \frac{\pi}{2^{(M-1)}} \right] \quad (4.14)$$

In der LUT ist die Sinus-Funktion entsprechend normiert auf die Amplitudenauflösung nach

$$f'(x) = \text{round}((2^{P-1} - 1) \sin(2\pi \frac{x}{2^M})) \quad (4.15)$$

für den Wertebereich $0 \leq x \leq 2^{M-2} - 1$ abgelegt. Die Bestimmung des Quadranten kann durch diese Art der Kodierung über die oberen zwei Bit erfolgen. Tabelle 4.2 gibt Aufschluss über die Bestimmung des Quadranten (Q) und die erforderliche Modifikation des Argumentes bzw. des Ergebnisses der LUT zur Erhaltung der 4-Quadranten Sinus-Funktion $f(x)$.

x(M-1...M-2)	Q	Wertebereich	LUT Modifikation
0	0	I $0 \leq x \leq 2^{M-2} - 1$	$f(x) = f'(x)$
0	1	II $2^{M-2} \leq x \leq 2^{M-1} - 1$	$f(x) = f'(2^{M-2} - x)$
1	0	III $2^{M-1} \leq x \leq 2^{M-1} - 2^{M-2} - 1$	$f(x) = -f'(x)$
1	1	IV $2^{M-1} - 2^{M-2} \leq x \leq 2^M - 1$	$f(x) = -f'(2^{M-2} - x)$

Tabelle 4.2: Bestimmung der Sinus-Quadranten

Für die Erzeugung des LUT-Inhaltes wurden verschiedene Matlab[©] Funktionen erstellt. Die Funktion `generate_vhdl_lut()` erzeugt VHDL-Quellcode für beliebige LUT Inhalte. Es wurde darauf geachtet, dass die Syntax vom Altera[©] Synthesewerkzeug Quartus[©] II als LUT erkannt und als *embedded RAM* synthetisiert wird. Die benötigte Sinus-Funktion ist in `sine_lut.m` definiert und wird als *function handle* übergeben. Das Programm `sine_lut_gen.m` erzeugt daraus LUT-Tabellen mit verschiedenen Wortbreiten, damit diese später einfach ausgetauscht werden können.

4.8.3 PI-Regler

Die Implementierung des zeitdiskreten PI-Reglers stellt eine Umsetzung der Übertragungsfunktion (2.63) auf Seite 28 dar. Die Differenzgleichung dieser Übertragungsfunktion mit $a_1 = -1$ lautet

$$y(n) = b_0x(n) + b_1x(n - 1) + y(n - 1) \quad (4.16)$$

Die Struktur dieser Differenzgleichung zeigt Abbildung 4.15.

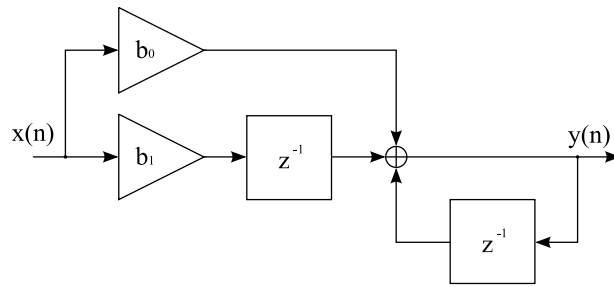


Abbildung 4.15: Struktur der Differenzgleichung des zeitdiskreten PI-Reglers

Da in dieser Form eine Multiplikation und eine Addition in einem Takt erfolgen muss, wurde die Übertragungsfunktion mit z^{-1} multipliziert. Dies ermöglicht die Einbringung von Pipeline-Registern. Das Resultat ist in Abbildung 4.16 zu sehen. So werden maximal zwei Additionen in einem Takt ausgeführt.

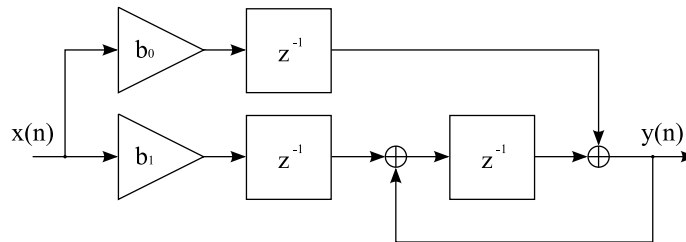


Abbildung 4.16: Blockschaltbild der Implementierung des diskreten PI-Reglers

Eine weitere Reduktion der Laufzeiten ist aufgrund der rekursiven Struktur problematisch, da ein Einfügen von Pipeline-Registern innerhalb der Rückkopplung zu einer Veränderung des Übertragungsverhaltens führt. In [MB04], [PM89a] und [PM89b] werden verschiedene Verfahren angegeben, die dieses Problem umgehen. Allerdings ist dies nicht ohne Mehraufwand möglich, so dass mindestens ein weiterer Multiplizierer nötig wird. Daher wurde auf die Implementierung einer solchen Methode verzichtet.

4.8.4 Tiefpassfilter

Zur Filterung der hochfrequenten Phasenfehler ist man nicht auf ein Tiefpassfilter mit linearem Phasengang angewiesen. Daher kann ein IIR-Filter verwendet werden. Dies wurde durch die Verwendung mehrerer zeitdiskreter PT1-Glieder erreicht, da diese einfach zu realisieren sind. Ein PT1-Glied stellt ein analoges Filter 1. Ordnung dar. Die Laplace-Übertragungsfunktion lautet

$$F(s) = \frac{1}{1 + sT} \quad (4.17)$$

wobei $f = 1/T$ die 3-dB Grenzfrequenz des Filters definiert. Das Filter hat ab dieser Grenzfrequenz eine asymptotische Dämpfung von 20dB pro Dekade. Die Anwendung der Rechteckregel (2.48) auf (4.17) liefert

$$F(z) = \frac{b_0}{1 + a_1 z^{-1}} \quad (4.18)$$

mit den Koeffizienten

$$b_0 = \frac{1}{1 + T/T_s} \quad (4.19)$$

und

$$a_1 = -\frac{1}{1 + T_s/T} \quad (4.20)$$

Für die Realisierung des resultierenden IIR-Filters werden zwei Multiplizierer und ein Addierer benötigt. Abbildung 4.17 zeigt die Struktur des Filters.

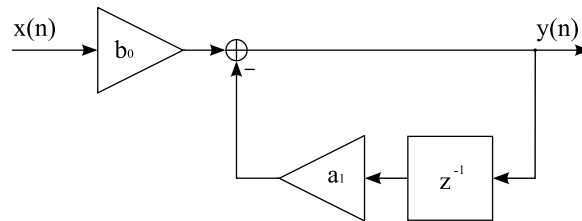


Abbildung 4.17: Aufbau des diskreten PT1-Gliedes

Eine wesentlich effizientere Implementierung ergibt sich, wenn man die Wahl der Grenzfrequenz einschränkt. Eine Multiplikation mit 2^N ($N \in \mathbb{Z}$) kann günstig durch eine Schiebeoperation realisiert werden. Der Ansatz

$$b_0 = \frac{1}{1 + T/T_s} \stackrel{!}{=} 2^{-N} \quad (4.21)$$

führt zu

$$a_1 = 2^{-N} - 1 \quad (4.22)$$

Dies kann mit einer Addition und einer Schiebeoperation statt mit einer Multiplikation realisiert werden. Die 3-dB Grenzfrequenz entspricht für diese Werte

$$f_g = \frac{f_s}{2^N - 1} \quad (4.23)$$

Besteht im Design die nötige Freiheit, die Grenzfrequenz nach (4.23) zu wählen, gelingt die Implementierung mit zwei Addierern und zwei Schiebeoperationen, die im FPGA lediglich eine „Umverdrahtung“ darstellen.

Als problematisch zeigt sich auch hier die Einbindung von Pipelineregistern innerhalb rekursiver Strukturen. In einer Struktur nach Abbildung 4.17 müssten innerhalb eines Datentaktes, zwei Multiplikationen und eine Addition ausgeführt werden. Eine Transponierung des bisherigen Ergebnisses liefert Abhilfe, da das ohnehin vorhandene Register (z^{-1}) als Pipelineregister des Addierers verwendet werden kann. Zur Transponierung müssen Ein- und Ausgang des Blockschaltbildes vertauscht und die Richtung aller Datenpfade umgekehrt werden [OS99]. Die resultierende Struktur, die letztendlich implementiert wurde, zeigt Abbildung 4.18. Über ein VHDL-generic lässt sich die Anzahl der hintereinander geschalteten Filter und somit die Ordnung bestimmen. Der damit erzielbare Frequenzgang ist in Abbildung 4.19 für $N = 4$ mit variierender Filterordnung O dargestellt.

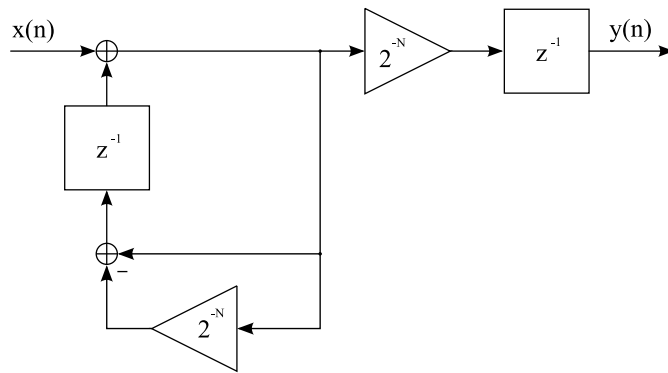


Abbildung 4.18: Aufbau des diskreten multipliziererfreien PT1-Gliedes

4.8.5 Phase Unwrap

In Abschnitt 2.7.4 wurde das Konzept des *Phase-Unwrapping* zur Erweiterung des Einfangbereichs der PLL vorgestellt. Die Kodierung der Phase nach (4.14) erlaubt eine effiziente Implementierung. Nach dieser hat das Bit an der Position i die Wertigkeit $\pi/2^{i-M+1}$, d.h. das MSB (Bit $M - 1$) hat die Wertigkeit π , Bit $M - 2$ hat die Wertigkeit $\pi/2$, etc. Um größere Phasen als π darstellen zu können, ist daher eine Erweiterung der Wortbreite und eine Kommaverschiebung notwendig. Jedes Bit zusätzliche Wortbreite verdoppelt den darstellbaren Phasenbereich und verdoppelt somit den Fangbereich der PLL. Der Aufbau der Implementierung ist in Abbildung 4.20 dargestellt.

Analog zu Abbildung 2.19 wird zunächst die Differenzphase gebildet. Da die Differenzphase Werte zwischen -2π und 2π annehmen kann, muss die Wortbreite um ein Bit erweitert werden (die Wortbreiten sind an den Verbindungslinien entsprechend gekennzeichnet). Um festzustellen, ob die Differenzphase größer oder kleiner als π ist,

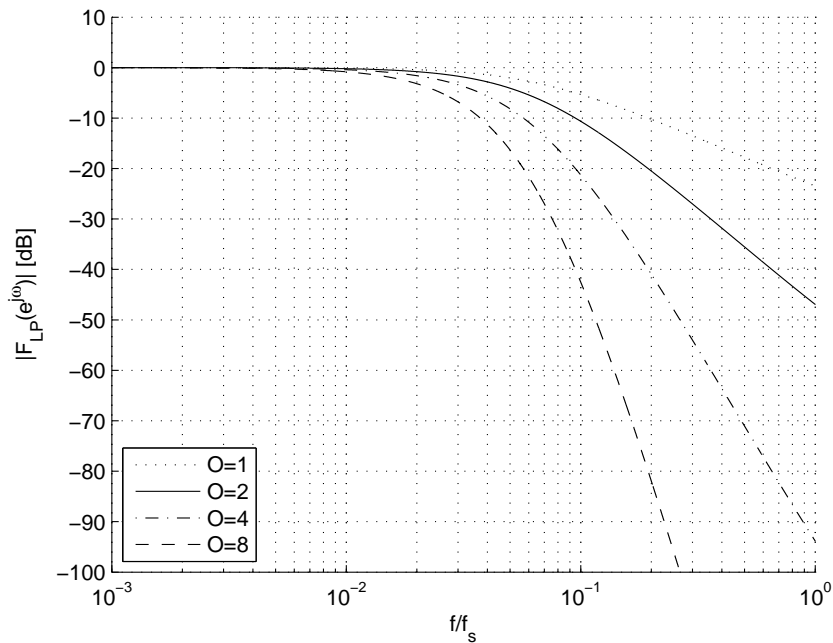


Abbildung 4.19: Betragsfrequenzgang des resultierenden Tiefpassfilters

reicht es aus, die oberen zwei Bit von `delta_phase` auszuwerten, um das Phaseninkrement (`phase_inc`) des Phasenakkumulators zu ermitteln. Tabelle 4.3 gibt die Wahrheitstabelle des Decoders (Block „DEC“) sowie den Wertebereich der Differenzphase wieder. Ist diese größer als π , muss 2π hinzuaddiert werden (`phase_inc=1`), ist sie kleiner als $-\pi$, müssen 2π abgezogen werden (`phase_inc=-1`). Ansonsten bleibt der Phasenakkumulators auf dem letzten Wert stehen (`phase_inc=0`).

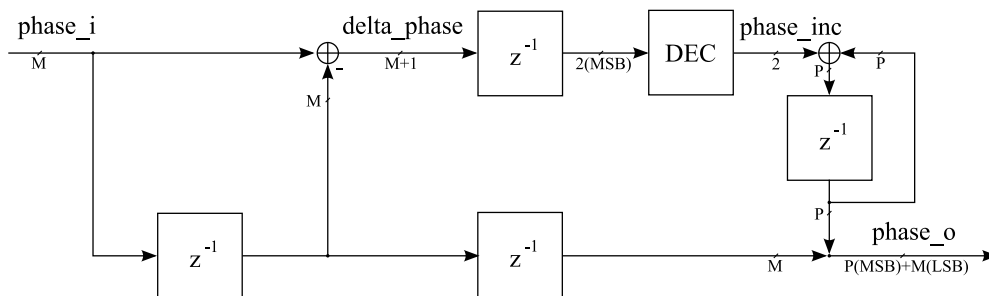


Abbildung 4.20: Aufbau der Phase-Unwrap Implementierung

delta_phase		phase_inc		Bereich von $\Delta\varphi$
1	0	1	0	
0	0	0	0	$0 \leq \Delta\varphi < \pi$
0	1	1	1	$\pi \leq \Delta\varphi < 2\pi$
1	0	0	1	$-2\pi \leq \Delta\varphi < -\pi$
1	1	0	0	$-\pi \leq \Delta\varphi < 0$

Tabelle 4.3: Auswertung der Phasendifferenz

Das Ergebnis des Phasenakkumulators muss der Eingangsphase hinzuaddiert werden. Da der Wertebereich ganzen Vielfachen von 2π entspricht, kann die Eingangsphase direkt um das Phasenakkumulator-Wort erweitert werden (Bit M hat die Wertigkeit 2π , Bit $M + 1$ die Wertigkeit 4π , etc.).

Hinter jedem Addierer befindet sich ein Pipelineregister. Beim Phasenakkumulator wurde dies durch Verschieben des Akkumulator-Registers erzielt. Dadurch ist das Ergebnis des Phasenakkumulators 2 Takte verzögert, so dass die Eingangsphase ebenfalls um 2 Takte verzögert werden muss. Hierbei wird das Register zur Bildung der Differenzphase mit verwendet.

4.9 Ressourcenbedarf und Verzögerungszeiten der Komponenten

Die Ergebnisse der Synthese der implementierten Komponenten sind in Tabelle 4.4 zusammengefasst. Es werden insgesamt etwa 90% der FPGA Logik-Ressourcen und etwa 58% des eingebetteten Speichers benötigt. Außerdem sind die Verzögerungszeiten in ganzen Takten eingetragen. Die für die PLL relevanten Verzögerungszeiten betragen $D = 28$ (unterer Teil der Tabelle). Das Verhältnis zwischen D und f_s/f_n beträgt ca. $4 \cdot 10^{-3}$ und liegt somit einen Faktor 2,5 unter dem Kriterium (2.83).

Komponente	LE-Bedarf	Speicherbedarf [Bit]	Verzögerung
Abtastratenerhöhung HF	790	0	62
Abtastratenerhöhung ZF	752	0	36
Analytisches Filter	358	0	7
Komplexe Multiplikation	1045	0	3
Phasendetektor (Cordic)	403	0	18
DDS	99	53248	3
PI-Regler	1694	0	2
Tiefpassfilter	191	0	3
Phase Unwrap	59	0	2
Summe	5391	53248	136

Tabelle 4.4: Syntheserergebnisse und Verzögerungszeiten der Komponenten

5 Simulation

Die Simulation der PLL-Implementierung nahm einen großen Teil der Bearbeitungszeit in Anspruch, da diese sehr detailliert durchgeführt werden musste. Im Gegensatz dazu nahm die eigentliche Implementierung weniger Zeit in Anspruch, da durch die Simulation die genaue Vorgehensweise klar definiert wurde. Die Simulation erfolgte Schritt für Schritt, von einem ersten zeitkontinuierlichen Modell, bis hin zur zeitdiskreten Simulation der ADPLL mit Frequenzoffset und Abtastratenerhöhung unter Berücksichtigung aller implementierten VHDL-Komponenten. Alle Simulationen verwenden die PLL-Bibliothek, die im folgenden Unterkapitel beschrieben werden soll. Die unterschiedlichen, teilweise aufeinander aufbauenden Simulationen können aus Platzgründen an dieser Stelle nicht ausführlich beschrieben werden. Tabelle 5.1 gibt daher einen Überblick über die realisierten Simulations-Modelle. Es soll nur die Simulation der zeitkontinuierlichen PLL (`continuous_ideal_pll`) sowie die Simulation des umfangreichsten Modells (`discrete_offset_oversampling_adpll_vhdl`) ausführlich beschrieben werden. Die Simulation der zeitkontinuierlichen PLL gibt Aufschluss über die nichtlinearen Effekte, die z.B. bei einem Frequenzsprung, der größer als die Ausrastfrequenz ist, auftreten. Im Modell `discrete_offset_oversampling_adpll_vhdl` wird die komplette ADPLL mit Frequenzoffset, Abtastratenerhöhung und allen implementierten VHDL-Komponenten simuliert.

5.1 PLL-Bibliothek für Matlab/Simulink

Für die Simulationen der PLL wurde eine PLL-Bibliothek für Matlab[®]/Simulink[®] entwickelt. Einfache Elemente, die öfters verwendet wurden (zum Beispiel ein VCO), sind mit herkömmlichen Simulink-Komponenten schwer realisierbar, da der Standard-Sinusgenerator von Simulink[®] (Sine Wave) die Einstellung der Frequenz zur Laufzeit nicht direkt zulässt. In Tabelle 5.2 sind die Komponenten zusammengefasst und beschrieben. Um die Bibliothek aus dem *Library Browser* zugänglich zu machen, muss der Pfad zum Verzeichnis `pll_lib` mittels `addpath(<Dateipfad>)` bekannt gemacht werden. Nach einem Neustart des *Library Browsers* erscheint die neue Bibliothek *Phase Locked Loop Components* und kann verwendet werden.

Name des Simulink-Modells	Beschreibung
<code>continuous_phase_model</code>	Simulation des zeitkontinuierlichen Phasenmodells zur Validierung der Übertragungsfunktionen (2.26) und (2.27).
<code>continuous_ideal_pll</code>	Simulation einer idealen zeitkontinuierlichen PLL zur Validierung des Phasenmodells und zur Simulation des Empfangsbereichs.
<code>continuous_ideal_offset_pll</code>	Simulation einer idealen zeitkontinuierlichen Offset-PLL nach der direkten Methode mit Addition der Phasensignale.
<code>discrete_phase_model</code>	Simulation des zeitdiskreten Phasenmodells mit DDS zur Validierung der Übertragungsfunktionen (2.69) und (2.70).
<code>discrete_ideal_pll</code>	Simulation einer idealen zeitdiskreten PLL zur Validierung des Phasenmodells und zur Simulation des Empfangsbereichs.
<code>discrete_phasedomain_pll</code>	Simulation der in Kapitel 3.2 beschriebenen Binärzähler-PLL-Architektur.
<code>discrete_adpll</code>	Simulation der in Kapitel 3.3 beschriebenen PLL-Architektur (ADPLL) ohne Frequenzoffset.
<code>discrete_offset_adpll</code>	Simulation der ADPLL mit Frequenzoffset.
<code>discrete_offset_upsampling_adpll</code>	Simulation der ADPLL mit Frequenzoffset und Abtastratenerhöhung.
<code>discrete_offset_upsampling_adpll_vhdl</code>	Simulation der ADPLL mit Frequenzoffset und Abtastratenerhöhung unter Berücksichtigung aller VHDL-Komponenten.

Tabelle 5.1: Realisierte Simulink-Modelle

<p>VCO</p>	<p>Analoger VCO: Frequenz und Phase können zur Laufzeit verändert werden. Der sin-Ausgang erzeugt ein um 90° verschobenes Signal gegenüber dem cos-Ausgang.</p>
<p>DDS</p>	<p>DDS: Die Wortbreite des FTW kann als Parameter vorgegeben werden. FTW und PTW können zur Laufzeit verändert werden und berechnen sich nach (2.55) und (2.56).</p>
<p>Phase Detector</p>	<p>Idealer Phasendetektor: Ermittelt die Phasendifferenz aus den beiden komplexen Signalen c1 und c2 und korrigiert dabei eventuell auftretende Überläufe.</p>
<p>Phase Difference</p>	<p>Ermittelt die Phasendifferenz aus zwei Phasensignalen und korrigiert dabei eventuell auftretende Überläufe.</p>
<p>Phase Sum</p>	<p>Ermittelt die Summe zweier Phasensignale und korrigiert dabei eventuell auftretende Überläufe.</p>
<p>Frequency Detector</p>	<p>Ermittelt die Frequenz eines Phasensignals über die diskrete Ableitung.</p>
<p>Phase Unwrap</p>	<p>Umsetzung der Phase-Unwrap Methode aus Kapitel 2.7.4.</p>
<p>AGC</p>	<p>Umsetzung einer automatischen Verstärkungsregelung (AGC) zur Kompensation von Pegeldifferenzen in einer PLL.</p>

Tabelle 5.2: Komponenten der Simulink[®] PLL-Bibliothek

5.2 Verifikation des zeitkontinuierlichen Phasenmodells

Im Folgenden sollen die Ergebnisse der Simulation der zeitkontinuierlichen PLL erläutert werden, die zur Verifikation des Phasenmodells sowie der Herleitung der Ausrastfrequenz verwendet wurde. Abbildung 5.1 zeigt den Aufbau des Simulink-Modells. Die Simulation wurde mit den PLL Parametern $\zeta = 0.707$ und $\omega_n = 2\pi rad/s$ gemacht. Die Konstanten des Reglers wurden entsprechend den Formeln (2.24) und (2.25) berechnet. Abbildung 5.2 zeigt den Phasenfehler bei einem Phasensprung des Referenzoszillators um 180° . Die numerisch ermittelte Sprungantwort nach Formel (2.27) ist dieser überlagert. Sie ist nahezu deckungsgleich und zeigt somit exemplarisch die Gültigkeit des Phasenmodells.

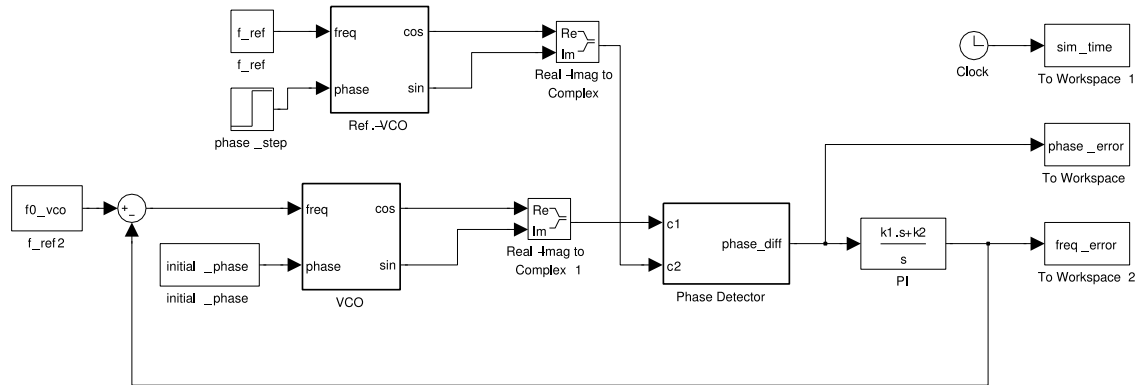


Abbildung 5.1: Simulink-Blockschaltbild der analogen PLL mit idealem Phasendetektor

Das Verhalten der PLL bei unterschiedlichen Frequenzsprüngen zeigt Abbildung 5.3. In Abbildung 5.3(a) ist das Verhalten des Phasenfehlers bei einem Frequenzsprung um 1 Hz dargestellt. Der Phasendetektor arbeitet im linearen Bereich und die Antwort auf eine Phasenrampe des Phasenmodells liefert das gleiche Ergebnis. Wie zu erwarten war, kann der Phasenfehler ausgeregelt werden. In Abbildung 5.3(b) ist das Einschwingen der Ausgangsfrequenz dargestellt. Nach Gleichung (2.90) ergibt sich für diese PLL eine Ausrastfrequenz von 6,89 Hz ($\varphi_{det,max} = 180^\circ$). Die Abbildungen 5.3(c) und 5.3(d) zeigen das Verhalten der PLL, wenn diese Frequenz überschritten wird. Auch bei dieser geringfügigen Abweichung von ca. 0,1 Hz benötigt die PLL wesentlich länger zum Einrasten. Zum Vergleich werden bei einem Frequenzoffset von 20 Hz, 7 Sekunden zum Einrasten benötigt, bei 60 Hz beträgt die Zeit 60 Sekunden. Die maximale Pull-In-Frequenz wurde experimentell zu 63 Hz ermittelt - über dieser Frequenzdifferenz ist die PLL instabil.

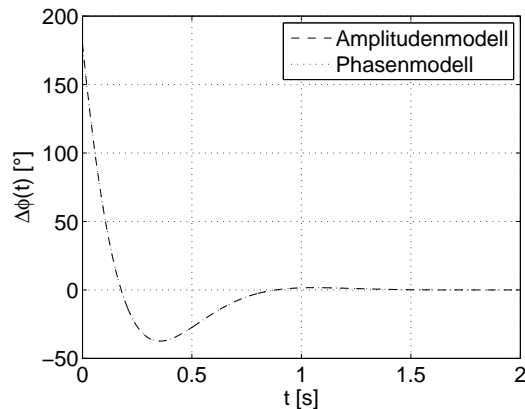
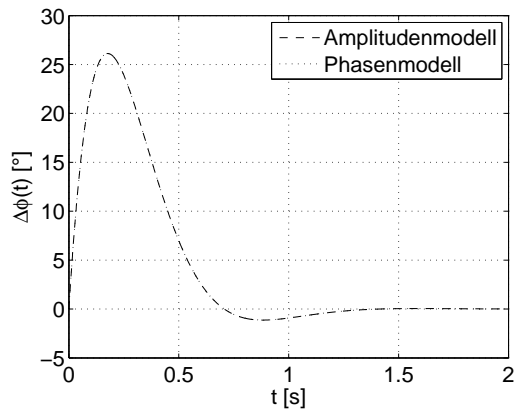


Abbildung 5.2: Phasenfehler bei einem Phasensprung von 180°

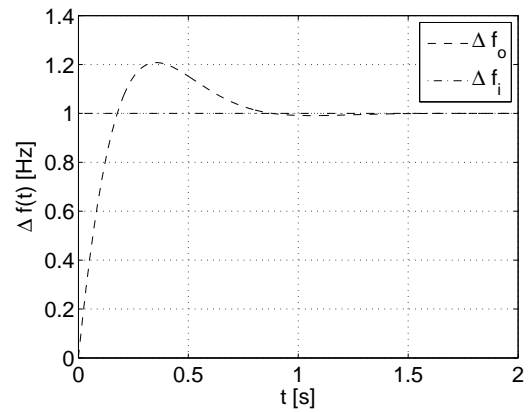
5.3 Simulation der Offset-ADPLL mit implementierten VHDL Komponenten

In diesem Abschnitt soll das komplexeste Modell beschrieben werden (`discrete_offset_upsampling_adpll_vhdl`). In Abbildung 5.4 ist das Simulink-Blockschaltbild der obersten Ebene zu sehen. In diesem sind alle Komponenten als Simulink-Modell sowie als VHDL-Implementierung realisiert. Welches Modell der jeweiligen Komponente verwendet werden soll, kann über Schalter aus einem Matlab-Skript bestimmt werden. Somit war es möglich, direkt zu unterscheiden, ob durch die Implementierung zusätzliche Fehler (z.B. durch Quantisierungsfehler) verursacht werden. Außerdem kann aus dem gleichen Matlab-Skript zwischen unterschiedlichen Simulink-Modellen ausgewählt werden. Da die Simulation mit VHDL-Komponenten erheblich länger dauert als ohne, kann somit das PLL-Verhalten in unter einer Minute simuliert werden, um bei Erfolg die Simulation mit VHDL-Komponenten zu wiederholen, die einige Minuten benötigt.

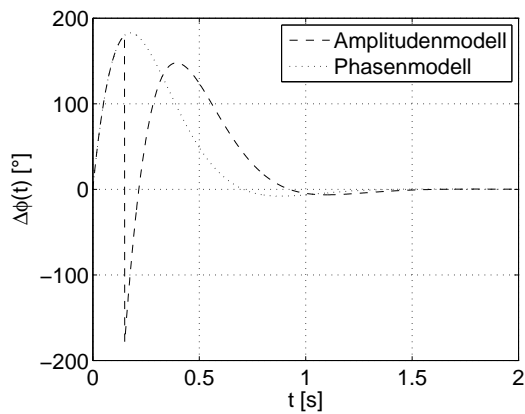
Es wurde der 'FixedStep'-Solver von Simulink[©] verwendet, da die komplette PLL fast ausschließlich zeitdiskrete Komponenten enthält. Lediglich die Ausgangs-Tiefpassfilter („7th Order Nyquist LP“) sind zeitkontinuierlich. Der 'FixedStep'-Solver arbeitet mit einer zeitlich äquidistanten Diskretisierung. Um das analoge Verhalten der Tiefpassfilter nachbilden zu können, kann die Diskretisierung auf ein Vielfaches der PLL-Abtastperiode (120 MHz) gesetzt werden. Für das prinzipielle Verhalten der PLL ist dies jedoch nicht notwendig.



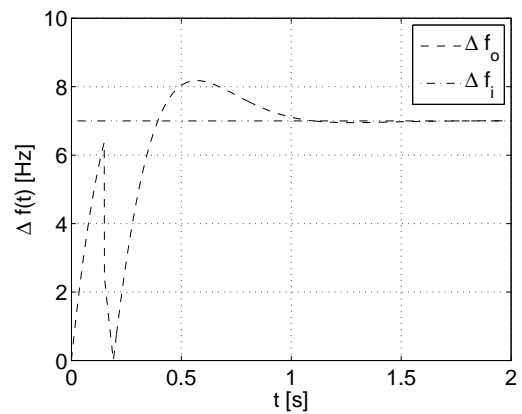
(a) Phasenfehler bei Frequenzsprung 1 Hz



(b) Änderung der Ausgangsfrequenz bei Frequenzsprung 1 Hz



(c) Phasenfehler bei Frequenzsprung 7 Hz



(d) Änderung der Ausgangsfrequenz bei Frequenzsprung 7 Hz

Abbildung 5.3: Phasenfehler und Ausgangsfrequenz bei verschiedenen Eingangsfrequenzsprüngen

5.3 Simulation der Offset-ADPLL mit implementierten VHDL Komponenten

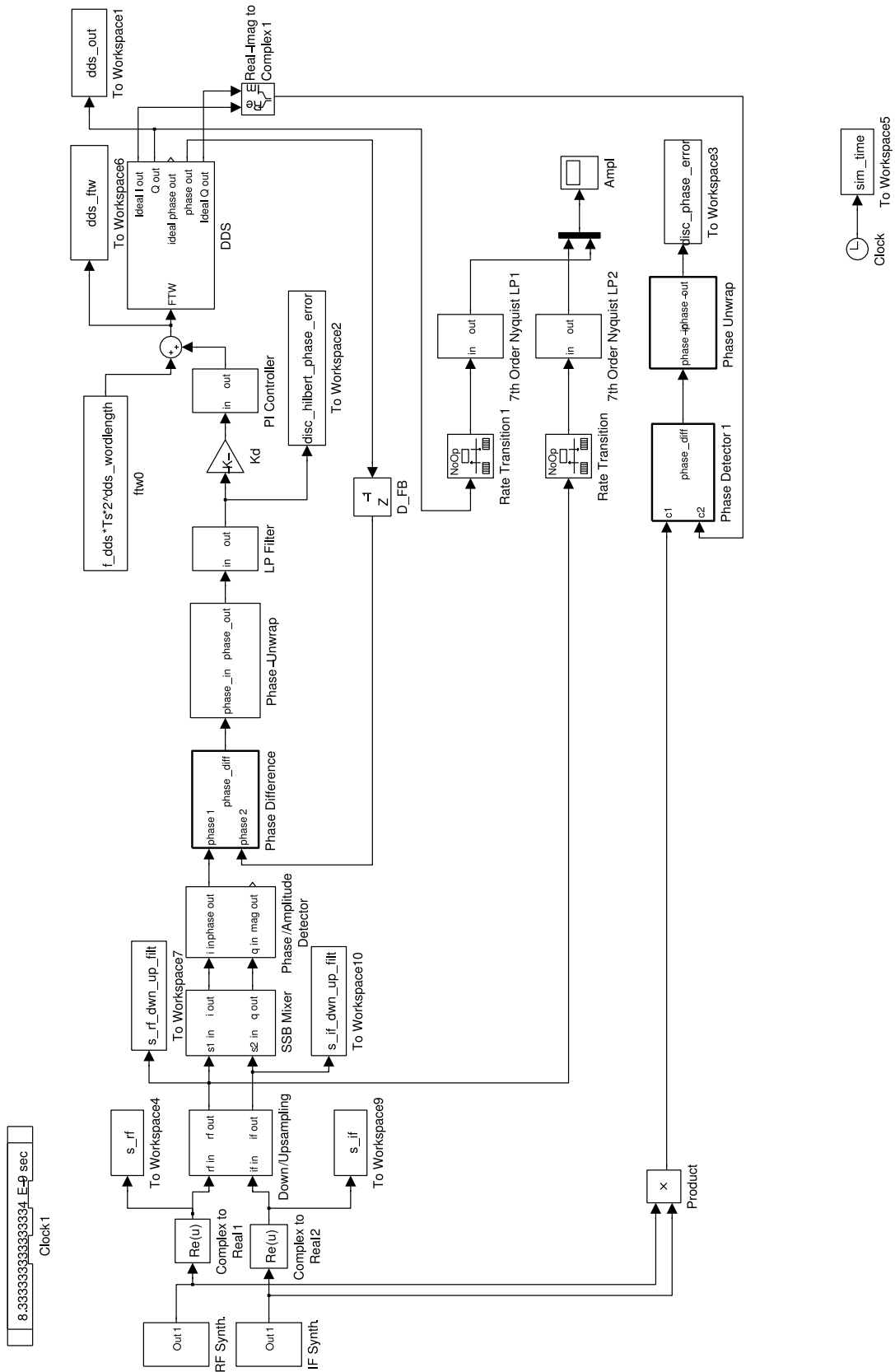


Abbildung 5.4: Simulink-Modell der Offset-ADPLL

Das Simulink-Modell enthält einige Subsysteme, die hier aus Platzgründen nicht dargestellt werden können. Sie werden daher im folgenden in ihrer Funktion beschrieben. Die Blöcke „RF Synth“ und „IF Synth“ sind Subsysteme, die die Eingangssignale der PLL mit DDS-Generatoren aus der PLL-Bibliothek erzeugen. Frequenz und Phase können hierbei über „Step“-Generatoren gesteuert werden, um Phasen- und Frequenzsprünge als Testsignal zu erzeugen. Sie erzeugen komplexe Signale, die über die Komponenten „Complex to Real“ in reelle Signale konvertiert werden. Das nachfolgende Subsystem „Down/Upsampling“ simuliert die Unterabtastung der ADCs von 30 MHz und die darauf folgende Abtastratenerhöhung (siehe Kapitel 4.5). Die Abbildungen 5.5 und 5.6 zeigen die Simulationsergebnisse für ein HF-Signal von 1 MHz sowie für das ZF-Signal mit 21,4 MHz. Es wurde eine FFT der Ein- und Ausgangssignale mit je 2^{15} Punkten berechnet, die zuvor mit einer Hamming-Fensterfunktion gewichtet wurden. Das HF-Signal wird besser interpoliert, als es die FFT auflösen kann. Beim ZF-Signal ist die Unterdrückung der Spiegelfrequenzen besser als 65 dB.

Darauf folgt ein Subsystem „SSB-Mixer“, der zwei analytische Filter (Kapitel 4.6) sowie einen komplexen Multiplizierer (Kapitel 4.7) zur Erzeugung des Offset-Signals enthält. Das analytische Filter ist im idealen Fall durch den Matlab-Block „FDA-Tool“ realisiert, das mit der gleichen Impulsantwort wie das verwendete Hilbert-Filter konfiguriert wurde. Im darauf folgenden Subsystem „Phase/Amplitude Detector“ ist der Phasendetektor enthalten, der im Falle der idealen Komponente mit dem Simulink-Block „Complex to Angle“ realisiert wird und ansonsten die Cordic-VHDL-Komponente enthält (Kapitel 4.8.1). Der Block „Phase Difference“ wurde aus der PLL-Bibliothek verwendet. Im Subsystem „Phase-Unwrap“ befindet sich die gleichnamige Komponente aus der PLL-Bibliothek für den idealen Fall, ansonsten die entsprechende VHDL-Implementierung (Kapitel 4.8.5). Das Gleiche gilt für den darauf folgenden Tiefpass „LP Filter“ und „PI Controller“ (Kapitel 4.8.4 und 4.8.3), die im idealen Fall durch ihre zeitdiskrete Übertragungsfunktion modelliert sind. Im Subsystem „DDS“ befindet sich im idealen Fall ein DDS-Generator aus der PLL-Bibliothek, der nach außen geführt ist, um eine exakte Phasenmessung durchführen zu können. Ansonsten befindet sich noch das Modell der VHDL-Implementierung in diesem Subsystem. Die Phase der DDS wird auf den „Phase Difference“ Block zurückgeführt um den Regelkreis zu schliessen. Die Blöcke „To Workspace“ dienen zur weiteren Verarbeitung der Signale in Matlab[©].

Die Parameter der PLL wurden größtenteils mit Hilfe der Formeln aus Kapitel 2 berechnet. Lediglich die Grenzfrequenz des Tiefpassfilters zur Unterdrückung der hochfrequenten Phasenfehler wurde durch die Simulation ermittelt. Dabei musste zwischen Störunterdrückung und Beeinflussung der PLL-Übertragungsfunktion abgewägt werden. Eine Grenzfrequenz von etwa 945 kHz lieferte einen guten Kompromiss. Tabelle 5.3 fasst alle PLL-Parameter zusammen. Im oberen Teil der Tabelle sind die vorgegebenen Parameter wiedergegeben, im unteren Teil finden sich die daraus resultierenden Eigenschaften wieder.

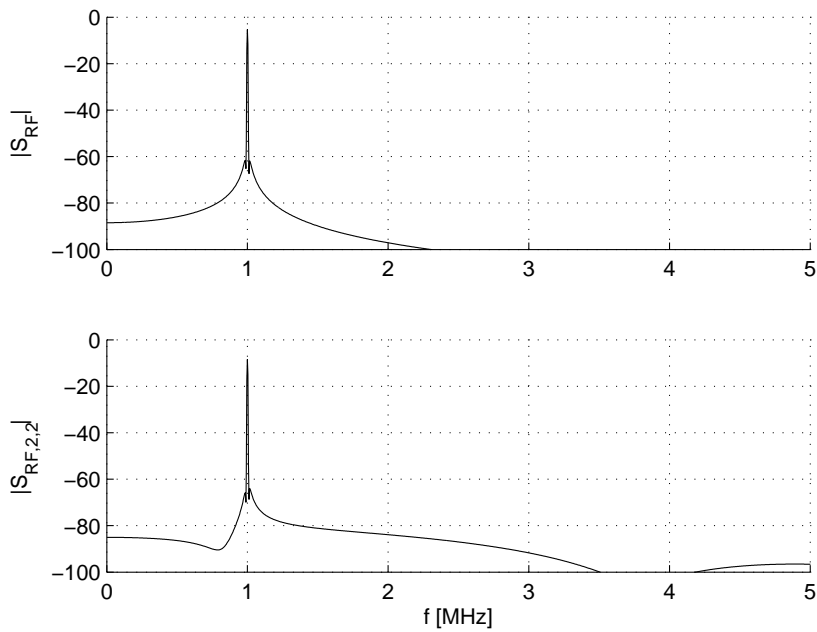


Abbildung 5.5: FFT-Spektrum vor und nach Abtastratenerhöhung des HF-Signals

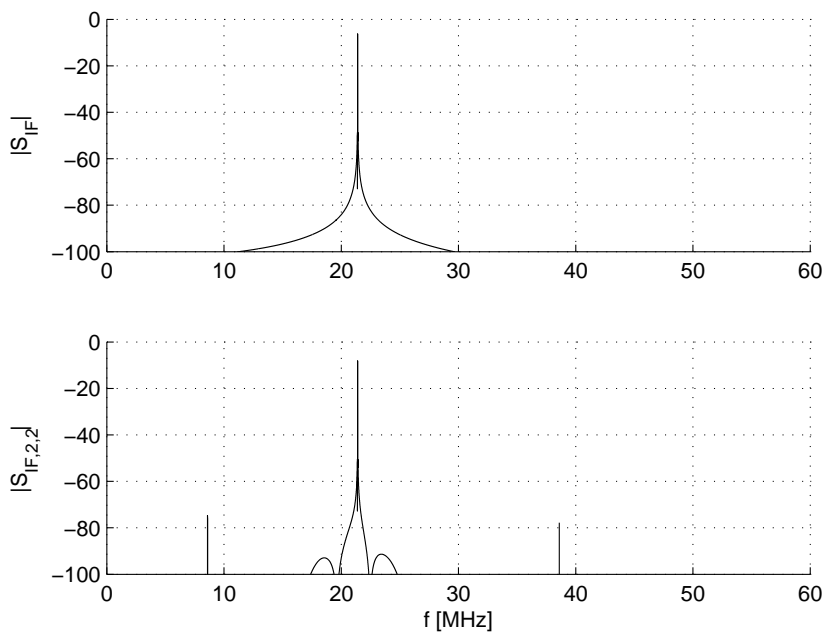


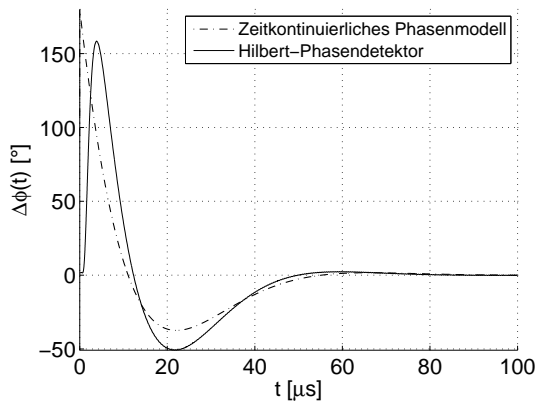
Abbildung 5.6: FFT-Spektrum vor und nach Abtastratenerhöhung des ZF-Signals

Parameter	Symbol	Wert
Abtastfrequenz	f_s	120 MHz
Eigenfrequenz	f_n	16 kHz
Dämpfung	ζ	0,707
Grenzfrequenz Tiefpassfilter	f_g	$f_s/(2^7 - 1) = 944,881$ kHz
Tiefpassfilter-Ordnung	O	3
Phasenbereichs-Erweiterung	P	7 Bit (entsprechen $2^7 \cdot \pm 180^\circ = \pm 23040^\circ$)
Mittelfrequenz	f_c	23,7 MHz
Phasenwortbreite DDS	M	14 Bit
Frequenzwortbreite DDS	N	32 Bit
Regler Koeffizient 0	b_0	0,303435
Regler Koeffizient 1	b_1	-0,303255
Fangbereich	f_L	$f_c \pm 14$ MHz
maximale Fangzeit	T_L	100 μ s

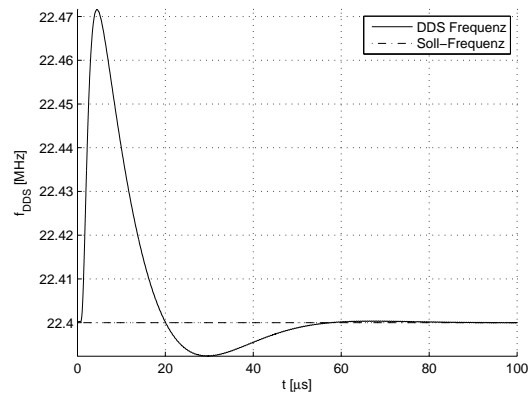
Tabelle 5.3: Zusammenfassung der PLL Parameter

Die Simulationsergebnisse für verschiedene Phasen- und Frequenzsprünge sind in Abbildung 5.7 dargestellt. Zum Vergleich ist die Sprungantwort der zeitkontinuierlichen Fehlerübertragungsfunktion für den Phasenfehler gestrichelt eingezeichnet (linke Seite). Bei der Darstellung der Ausgangsfrequenz (rechte Seite) ist die Soll-Frequenzvorgabe gestrichelt eingezeichnet. Die Frequenz des HF-Eingangssignals vor dem Frequenzsprung betrug 1 MHz. Man erkennt eine Abweichung des Phasenfehlers zum kontinuierlichen Modell. Dieser wird primär durch das Tiefpassfilter hervorgerufen, das nur im diskreten Fall wegen des Hilbert-Phasendetektors benötigt wird. Die Einfangzeit wird dadurch jedoch nicht beeinflusst. Wie die letzten vier Abbildungen 5.7(c) bis 5.7(f) zeigen, ist die Einfangzeit unabhängig von der Höhe des Frequenzsprungs, solange dieser innerhalb des Fangbereichs von ± 14 MHz liegt.

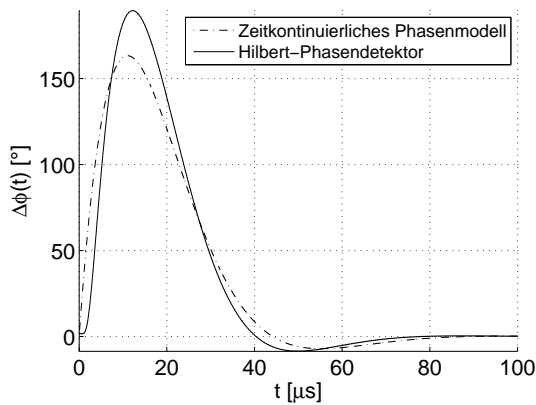
5.3 Simulation der Offset-ADPLL mit implementierten VHDL Komponenten



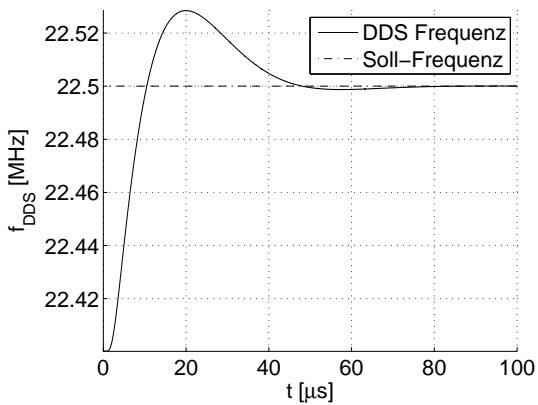
(a) Phasenfehler bei 180° Phasensprung



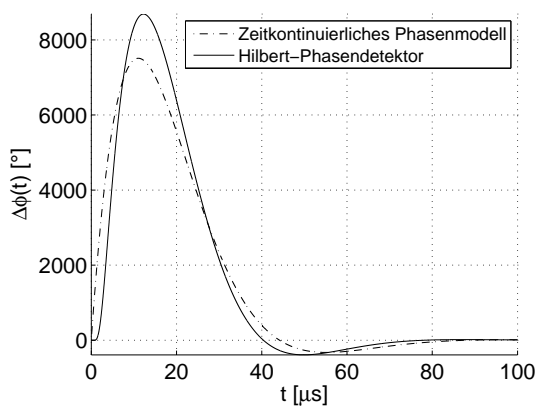
(b) Ausgangsfrequenz bei 180° Phasensprung



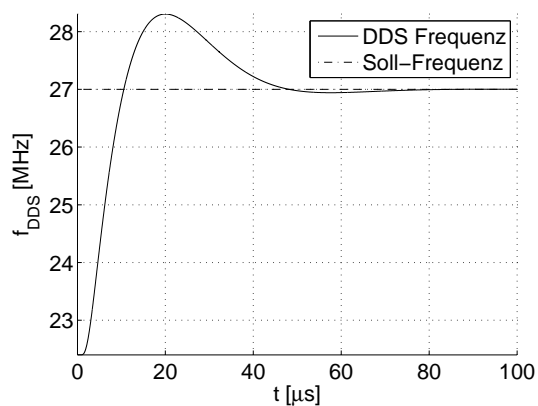
(c) Phasenfehler bei 100 kHz Frequenzsprung



(d) Ausgangsfrequenz bei 100 kHz Frequenzsprung



(e) Phasenfehler bei 4,6 MHz Frequenzsprung



(f) Ausgangsfrequenz bei 4,6 MHz Frequenzsprung

Abbildung 5.7: Simulationsergebnisse der Sprungantworten

6 Ergebnisse

6.1 Messergebnisse

Nachfolgend sollen die erzielten Messergebnisse am implementierten Offset-LO vorgestellt werden. Die Messungen konnten aus zeitlichen Gründen nicht an der endgültigen Implementierung vorgenommen werden. Sie wurden an einem Zwischenstand vorgenommen, in welchem die zweite Stufe der Abtastratenerhöhung noch nicht implementiert war. Der Offset-LO lief somit bei einer internen Abtastfrequenz von 60 MHz. Die Funktion des Offset-LO ist damit trotzdem gewährleistet, bei der höheren Abtastfrequenz sind jedoch bessere Messergebnisse zu erwarten.

6.1.1 Phasenauflösung des DSP-Systems

Es wurden zeitaufgelöste Phasenmessungen am bzw. mit dem Offset-LO durchgeführt. Zur Messung wurde das DSP-System verwendet, da es eine hohe Phasen- und Zeitauflösung hat. In einer ersten Messung wurde die Phasenauflösung des DSP-Systems bestimmt, um die Genauigkeit der darauf folgenden Messungen beurteilen zu können. Dazu wurde ein HF-Generator mit konstanter Frequenz verwendet, dessen Signal über einen 3dB-Splitter an beide Messeingänge des DSP-Systems geführt wurde (vgl. Abbildung 1.2 auf Seite 7). Die Phasenmessung sollte daher Null Grad ergeben. Die Frequenz wurde auf die Mittenfrequenz von 3,1 MHz eingestellt. Das LO-Signal beider Kanäle wurde durch einen zweiten HF-Generator mit fester Frequenz (24,5 MHz) vorgegeben. Über den Host-PC des DSP-Systems können die Messwerte mit einer Zeitauflösung von 665,9 ns über einen Messzeitraum von mehreren Sekunden aufgenommen werden. Aus der DSP-Host-Software lassen sich die Daten in Form von Textdateien (CSV-Format) exportieren und in Matlab[©] auswerten. Da für die weiteren Messungen nicht die absolute Phase, sondern der absolute Phasenfehler von Interesse war, wurde der Mittelwert der Messung herausgerechnet. Abbildung 6.1 zeigt das Ergebnis der Messung über eine Sekunde. Die Phasenauflösung beträgt mit diesem Aufbau etwa $\pm 1^\circ$.

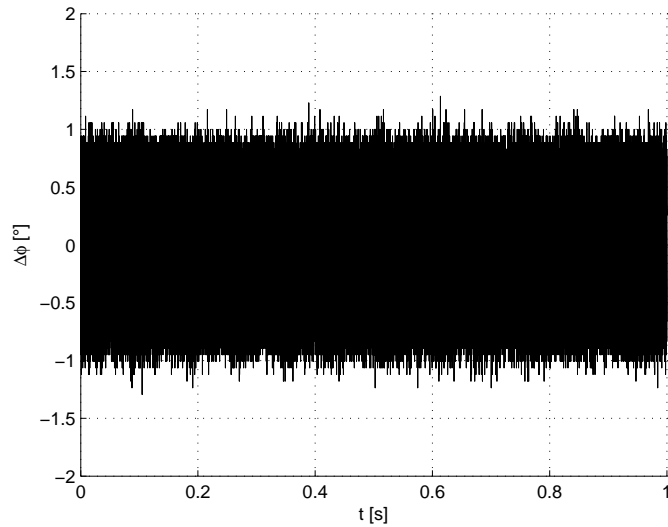


Abbildung 6.1: Phasenauflösung des DSP-Systems

6.1.2 Messung der Sprungantwort des Offset-LO

In dieser Messung sollte das zeitliche Verhalten des Offset-LO anhand der Simulationsergebnisse verifiziert werden. Da das Ausgangssignal des Offset-LO nicht im Frequenzbereich des DSP-Systems liegt, wurde das LO-Signal mit dem ZF-Signal gemischt (Mischer Minicircuits, SRA-1H) und anschließend tiefpassgefiltert, um das HF-Signal zurückzugewinnen. Dies trägt dafür weiterhin die Phaseninformation des LO-Signals. Der Messaufbau ist in Abbildung 6.2 dargestellt. Das bei der Mischung entstehende Spiegelsignal bei 45,9 MHz kann ohne Probleme durch das Tiefpassfilter beseitigt werden. Im eingeschwungenen Zustand entsteht lediglich ein konstanter Phasenoffset durch die unterschiedlichen Laufzeiten der Signale. Ein Phasensprung am HF-Generator liefert die gewünschte Sprungantwort des Phasenfehlers. Dieser wurde durch eine rechteckförmige Phasenmodulation des HF-Generators mit 180° und 100 Hz Periode realisiert. Das Ergebnis ist in Abbildung 6.3 zu sehen. In dieser ist ebenfalls das Ergebnis der Simulation dargestellt. Der bei der Messung unbekanntes Zeitpunkt des Phasensprungs wurde so verschoben, dass er deckungsgleich zur Simulation ist. Messung und Simulation zeigen selbst im absoluten Phasenfehler eine hervorragende Übereinstimmung.

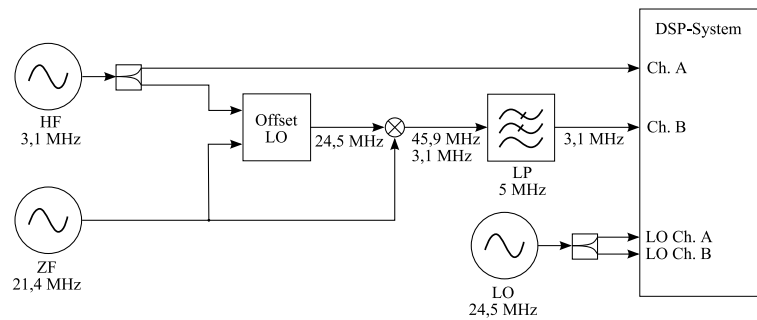
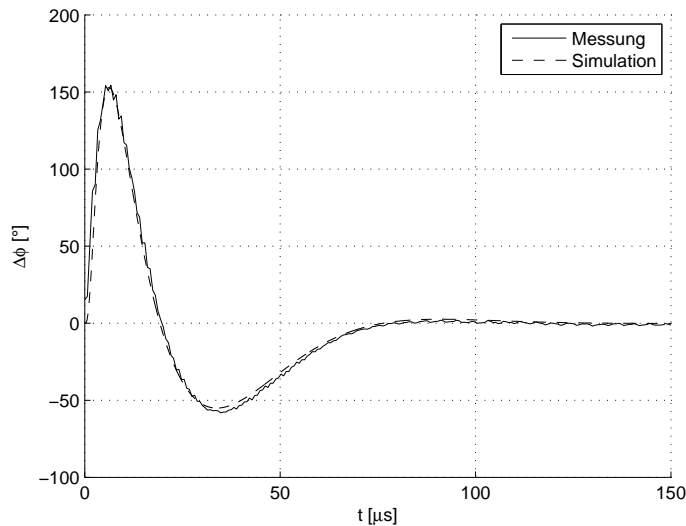


Abbildung 6.2: Messaufbau zur Bestimmung der Sprungantwort

Abbildung 6.3: Messung und Simulation eines Phasensprungs um 180°

6.1.3 Messung eines phasenmodulierten Signals

Als Nachweis der Funktion des Offset-LO in Zusammenarbeit mit dem DSP-System wurde die Messung eines bekannten Signals durchgeführt. Der analoge Offset-LO wurde durch den neuen Offset-LO in dieser Messung ersetzt. Zur Signalerzeugung dienten hierbei zwei HF-Signalgeneratoren, die über einen externen 10 MHz Takt miteinander synchronisiert wurden und jeweils an Kanal A und B des DSP-Systems angeschlossen wurden. Der Offset-LO wurde über einen 3 dB Splitter mit dem HF-Generator von Kanal A verbunden. Beide HF-Generatoren wurden auf eine feste Frequenz von 3,1 MHz eingestellt. Der Generator für Kanal B wurde mit einer sinusförmigen Phasenmodulation mit $\pm 90^\circ$ Phasenabweichung und 10 Hz Periode konfiguriert. Das Ergebnis

der Messung ist in Abbildung 6.4 zu sehen.

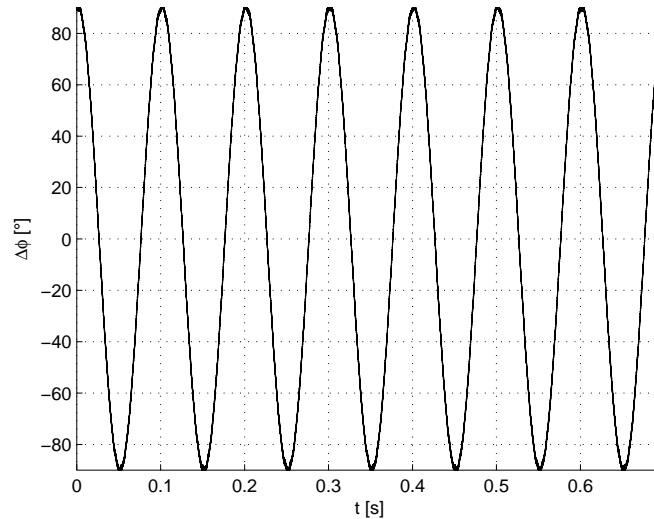


Abbildung 6.4: Messung eines phasenmodulierten Signals

6.1.4 Phasenfehler bei frequenzdynamischen Eingangssignalen

Die bisher vorgestellten Messungen erfolgten alle bei konstanter Frequenz. Um den Einfluss des Offset-LO bei einer Frequenzrampe zu ermitteln, wurde ein Aufbau ähnlich der Messung aus Abschnitt 6.1.1 gewählt. Diesmal wurde jedoch der HF-Generator in der Frequenz moduliert, so dass der Frequenzbereich von 0,8 MHz bis 5,4 MHz linear durchgeföhren wird. Außerdem kam statt einer konstanten LO-Frequenz der Offset-LO zum Einsatz. Die Messung wurde bei einer Modulationsfrequenz von 1 Hz (entsprechend $\dot{f}_{RF} = 4,6 \text{ MHz/s}$), sowie mit 17,5 Hz (entsprechend $\dot{f}_{RF} = 80,5 \text{ MHz/s} > \dot{f}_{RF,max}$) durchgeföhrt. Die Messergebnisse jeweils eines Frequenzdurchlaufs sind in Abbildung 6.5 dargestellt. Es zeigt sich, dass der Spitzenwert des Phasenfehlers nahezu unabhängig von der Modulationsfrequenz ist. Er weist dafür eine Frequenzabhängigkeit auf. Die Messauflösung des DSP-Systems über den gesamten Frequenzbereich verschlechtert sich um etwa $\pm 1^\circ$. Die resultierende Phasengenauigkeit erfüllt die in Tabelle 1.2 auf Seite 9 angegebenen Anforderungen. Wie Simulationen gezeigt haben, lässt sich das Phasenrauschen durch die Verwendung der höheren Abtastfrequenz noch weiter verbessern.

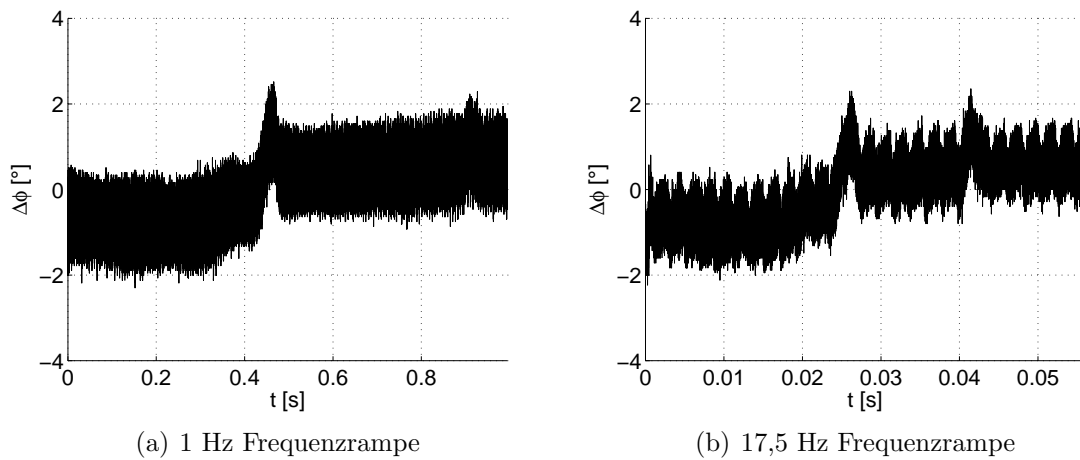


Abbildung 6.5: Messung des Phasenfehlers bei frequenzdynamischen Eingangssignalen

6.2 Ausblick

In dieser Arbeit konnte gezeigt werden, dass eine leistungsfähige Offset-LO Implementierung mit den zur Verfügung stehenden Komponenten möglich ist.

Die Bus-Schnittstelle zwischen FIB und ADC/DAC-FAB wurde als Engpass für die Abstraten der Wandler aufgedeckt. Eine Möglichkeit für eine Verbesserung wäre der Austausch des CPLD-Bausteins des ADC/DAC-FAB durch einen größeren FPGA. Damit könnte der Offset-LO auf dem ADC/DAC-FAB implementiert werden und wäre damit in der Lage, die volle Datenrate der Wandler zu verarbeiten. Das FIB hätte dabei eine rein steuernde Funktion.

Eine weitere Verbesserungsmöglichkeit wäre die Implementierung einer automatischen Verstärkungsregelung (AGC) zur Reduktion des Amplitudenfehlers der Hilberttransformation. In einer Simulation konnte gezeigt werden, dass die Verwendung einer einfachen AGC die Phasengenauigkeit des Offset-LO weiter verbessert. Dies führt jedoch zu einer komplexeren Dynamik des Gesamtsystems.

Literaturverzeichnis

- [AAHAQ06] AL-ARAJI, Saleh R. ; HUSSAIN, Zahir M. ; AL-QUTAYRI, Mahmoud A.: *Digital Phase Lock Loops*. Dordrecht : Springer, 2006. – ISBN 0–3873–2863–7
- [Alt06] ALTERA: *MAX II Device Handbook*. Altera, Dezember 2006
- [Alt07] ALTERA: *Cyclone Device Handbook*. Altera, Januar 2007
- [Ana06] Analog Devices: *Datasheet AD9854 - CMOS 300 MSPS Quadrature Complete DDS*. 2006
- [Bed62] BEDROSIAN: *The Analytic Signal Representation of Modulated Waveforms*. 1962
- [Bes93] BEST, Roland: *Theorie und Anwendungen des Phase-locked Loops*, 5. Aufl. Berlin, Offenbach : VDE Verlag, 1993. – ISBN 3–8007–1980–0
- [BLM04] BARRY, John R. ; LEE, Edward A. ; MESSERSCHMITT, David G.: *Digital Communication*. New York : Kluwer Academic Press, 2004. – ISBN 0–7923–7548–3
- [CR83] CROCHIERE, Ronald E. ; RABINER, Lawrence R. ; OPPENHEIM, Alan V. (Hrsg.): *Multirate Digital Signal Processing*. New York : Prentice Hall, 1983. – ISBN 0–1360–5162–6
- [Elf05] ELFIMOW, Alexander: *Realisierung eines DDS-HF-Synthesizers für digitale Regelungssysteme*, Fachhochschule Fulda, Diplomarbeit, Mai 2005
- [Fli91] FLIEGE, Norbert: *Systemtheorie*. Stuttgart : Teubner, 1991. – ISBN 3–519–06140–6
- [Fli93] FLIEGE, Norbert: *Multiraten-Signalverarbeitung: Theorie und Anwendungen*. Stuttgart : Teubner, 1993. – ISBN 3–519–06155–4
- [Fra87] FRAN CZAK, B. *SIS Parameter List*. 1987
- [G+06] GUTBROD, H. H. [u. a.]: *FAIR Baseline Technical Report*. Darmstadt, September 2006. – ISBN 3–9811298–0–6
- [Gar79] GARDNER, Floyd M.: *Phaselock Techniques, second edition*. New York, Chinchester, Brisbane, Toronto : John Wiley & Sons, Inc., 1979

- [Gar05] GARDNER, Floyd M.: *Phase-lock Techniques, third edition*. New York, Chinchester, Brisbane, Toronto : John Wiley & Sons, Inc., 2005
- [GSI07] GSI. *Internetseite der GSI* <http://www.gsi.de/> [02.07.2007]. 2007
- [GZMGK06] GUNTORO, Andre ; ZIPF, Dr.-Ing. P. ; H.C. MULT. MANFRED GLENER, Prof. Dr. D. ; KLINGBEIL, Dr.-Ing. H.: Statusbericht zum GSI-Projekt „Rekonfigurierbare Rechensysteme für digitale Hochfrequenzregelungen bei Schwerionenbeschleunigern“ - Extending the Phase Detector module: Magnitude and Phase Information / Technische Universität Darmstadt Fachgebiet Mikroelektronische Systeme. 2006. – Forschungsbericht
- [IB05] I.N. BRONŠTEIN, K.A.Semendjajew: *Taschenbuch der Mathematik*. Frankfurt am Main : Deutsch, 2005. – ISBN 3–8171–2016–8
- [Kam04] KAMMEYER, Karl D.: *Nachrichtenübertragung, 3. Aufl.* Stuttgart : Teubner, 2004. – ISBN 3–519–26142–1, 978–3–519–26142–1
- [KK06] KAMMEYER, Karl D. ; KORSCHER, Kristian: *Digitale Signalverarbeitung : Filterung und Spektralanalyse mit MATLAB-Übungen ; mit 33 Tabellen*. Wiesbaden : Teubner, 2006. – ISBN 3–8351–0072–6
- [Kli05] KLINGBEIL, Harald: A fast DSP-based phase-detector for closed-loop RF control in synchrotrons. In: *IEEE Transactions on Instrumentation and Measurement* 54 (2005), S. 1209–1213
- [Kro98] KROUPA, Věnceslav F.: *Direct Digital Frequency Synthesizers*. Piscataway, NJ : IEEE Press, 1998. – ISBN 0–7803–3438–8
- [Kro03] KROUPA, Věnceslav F.: *Phase Lock Loops*. Chichester, England : John Wiley & Sons Ltd, 2003
- [Kum06a] KUMM, Martin: *FIB-DDS Dokumentation*. GSI, Februar 2006
- [Kum06b] KUMM, Martin: *User Manual FPGA Interface Board 2*. 0.6. GSI, Juni 2006
- [MB95] MEYER-BÄSE, Uwe: *Der Einsatz komplexer Algorithmen zur Realisierung universeller Abtastempfänger in FPGA-Technik*, Technische Hochschule Darmstadt, Diss., 1995
- [MB04] MEYER-BÄSE, Uwe: *Digital Signal Processing with Field Programmable Gate Arrays, 2nd Edition*. Berlin : Springer, 2004. – ISBN 3–540–21119–5
- [Meh83] MEHRGARDT, Soenke: Noise Spectra of Digital Sine-Generators Using

- the Table-Lookup Method. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* ASSP-31 (1983), S. 68–70
- [Mic99] MICHAEL, T.: *Digitale Phasenregelkreise in Trägerregenerationsverfahren*, Universität Ulm, Diss., 1999
- [MW05] MORITZ, Peter ; WALTER, Horst: *Schaltplan DDS-FAB*. GSI/Ing.-Büro Walter, 2005
- [NS87] NICHOLAS, H.T. ; SAMUELI, H.: An Analysis of the Output Spectrum of Direct Digital Frequency Synthesizers in the Presence of Phase-Accumulator Truncation. In: *Proceedings of the 41st Annual Frequency Control Symposium*, 1987, S. 495–502
- [NS91] NICHOLAS, H.T. ; SAMUELI, H.: A 150-MHz direct digital frequency synthesizer in 1.25- μ m CMOS with -90-dBc spurious performance. In: *Solid-State Circuits, IEEE Journal of* 26 (1991), Dec., Nr. 12, S. 1959–1969
- [NSK88] NICHOLAS, H.T. ; SAMUELI, H. ; KIM, Bruce: The Optimization of Direct Digital Frequency Synthesizer Performance in the Presence of Finite Word Length Effects. In: *Proceedings of the 42nd Annual Frequency Control Symposium* (1988), S. 357–363
- [OM91] O’LEARY, Paul ; MALOBERTI, Franco: A Direct-Digital Synthesizer with Improved Spectral Performance. In: *IEEE Transactions on Communications* 39 (1991), S. 1046–1048
- [OS99] OPPENHEIM, Alan V. ; SCHAFER, Ronald W.: *Zeitdiskrete Signalverarbeitung*. München : Oldenbourg, 1999. – ISBN 3–486–24145–1
- [PM89a] PARHI, Keshab K. ; MESSERSCHMITT, David G.: Pipeline Interleaving and Parallelism in Recursive Digital Filters-Part I: Pipelining Using Scattered Look-Ahead and Decomposition. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 37 (1989), S. 1099–1117
- [PM89b] PARHI, Keshab K. ; MESSERSCHMITT, David G.: Pipeline Interleaving and Parallelism in Recursive Digital Filters-Part II: Pipelined Incremental Block Filtering. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 37 (1989), S. 1118–1134
- [RFB94] REILLY, Andrew ; FRAZER, Gordon ; BOASHASH, Boualem: Analytic Signal Generation - Tips and Traps. In: *IEEE Transactions on Signal Processing* 42 (1994), Nr. 11, S. 3241–3245
- [San06] SANJARI, Mohammad S.: *Hardware and Software Implementation of a*

- Radio Frequency High-Speed Data Conversion Unit for Digital Control Systems*, TU Darmstadt, Bachelorthesis, Oktober 2006
- [SB05] STASZEWSKI, R. B. ; BALSARA, P. T.: Phase-Domain All-Digital Phase-Locked Loop. In: *IEEE Transactions on Circuits and Systems* 52 (2005), S. 159–163
- [SSS05] SHU, Keliu ; SÀNCHEZ-SINENCIO, Edgar: *CMOS PLL Synthesizers: Analysis and Design*. Boston, MA : Springer, 2005. – ISBN 0–387–23669–4
- [Ste97] STENSBY, John L.: *Phase-Locked Loops - Theory and Applications*. Boca Raton, Fla : CRC Press, 1997. – ISBN 0–8493–9471–6
- [Ste02] STEPHENS, Donald R.: *Phase-Locked Loops for Wireless Communications*. Dordrecht : Kluwer Academic Press, 2002. – ISBN 0–7923–8204–8
- [Tay83] TAYLOR, Fred J.: *Digital Filter Design Handbook*. New York : Dekker, 1983. – ISBN 0–8247–1357–5
- [TS02] TIETZE, Ulrich ; SCHENK, Christoph: *Halbleiter - Schaltungstechnik 12. Auflage*. Berlin : Springer, 2002. – ISBN 3–540–42849–6
- [Unb00] UNBEHAUEN, Heinz: *Regelungstechnik II*. Braunschweig/Wiesbaden : Vieweg, 2000. – ISBN 3–528–73348–9
- [Unb05] UNBEHAUEN, Heinz: *Regelungstechnik I*. Wiesbaden : Vieweg, 2005. – ISBN 3–528–21332–9
- [Vol59] VOLDER, J. E.: The CORDIC Trigonometric Computing Technique. In: *IRE Transactions on Electronic Computers* (1959), S. 330–334
- [Wal71] WALTHER, J. S.: A unified algorithm for elementary functions. In: *Spring Joint Computer Conference*, 1971, S. 379–385
- [Win05] WINDIRSCH, Peter. *Skriptum der Vorlesung „VLSI Entwurf für die digitale Echtzeit-Signalverarbeitung“*; Technische Universität Darmstadt. 2005

7 Abkürzungen

ADC	Analog to D igital C onverter (dt. Analog/Digital-Umsetzer)
AGC	A utomatic G ain C ontrol (dt. automatische Verstärkungsregelung)
AM	A mplituden m odulation
BPSK	B inary P hase S hift K eying (dt. binäre Phasenumtastung)
CBM	C ompressed B aryonic M atter
CMOS	C omplementary M etal O xide S emiconductor
CORDIC	C oordinate R otation D igital C omputer
CPLD	C omplex P rogrammable L ogic D evice
CR	C ollector R ing
CSV	C omma S aperated V alues
DAC	D igital to A nalog C onverter (dt. Digital/Analog-Umsetzer)
DDS	D irect D igital S ynthesis
DSP	D igitaler S ignal p rozessor
ESR	E xperimentier- S peicherring
FAB	F IB A dapter B oard
FAIR	F acility for A ntiproton and I on R esearch
FFT	F ast F ourier T ransformation (dt. schnelle Fouriertransformation)
FIB	F PGA I nterface B oard
FIFO	F irst I n F irst O ut
FIR	F inite I mpulse R esponse
FLAIR	F acility for low-energy A ntiproton and I on R esearch
FPGA	F ield P rogrammable G ate A rray

FSF	F requency S ampling F ilter
FSK	F requency S hift K eying (dt. Frequenzumtastung)
FTW	F requency T uning W ord (Frequenzwort einer DDS)
GSI	G esellschaft für S chwerionen f orschung
HESR	H igh Energy E xperimental S torage R ing
HF	H och f requenz
IC	I ntegrated C ircuit
IF	I ntermediate F requency (dt. Zwischenfrequenz, ZF)
IIR	I nfinite I mpulse R esponse
LE	L ogic E lement
LSB	L east S ignificant B it
LUT	L ook U p T able (dt. Nachschlagetabelle)
LVTTL	L ow V oltage T ransistor- T ransistor L ogic
MSB	M ost S ignificant B it
MSPS	M ega S amples P er S econd (dt. Millionen Abtastungen pro Sekunde)
NCO	N umber C ontrolled O szillator
NESR	N ew E xperimental S torage R ing
Offset-LO	O ffset- L ocal O scillator
PD	P hasen d etektor
PI-Regler	P roportional- I ntegral-Regler
p-LINAC	P rotonen- L inear A ccelerator
PLL	P hase l ocked L oop (dt. Phasenstarre Regelschleife)
PTW	P hase T uning W ord (Phasenwort einer DDS)
PWM	P uls w eiten m odulation
RESR	R ecycled E xperimental S torage R ing
RF	R adio F requency (dt. Hochfrequenz, HF)
SFDR	S purious F ree D ynamic R ange (dt. störungsfreier Dynamikbereich)
SIS	S chwerionen S ynchrotron

SNR	S ignal to N oise R atio (dt. Signal-Rausch-Verhältnis)
SSB	S ingle S ideband (dt. Einseitenband, ESB)
Super-FRS	Super- F ragment- S eparator
TTL	T ransistor- T ransistor L ogic
UNILAC	U niversal L inear A ccelerator
VCO	V oltage C ontrolled O scillator (dt. Spannungsgesteuerter Oszillator)
VHDL	V ery H igh S peed I ntegrated C ircuit H ardware D escription L anguage.
ZF	Z wischenfrequenz