

# Realisierung einer adaptiven parallelen Hardwarearchitektur für hochperformante OFDMA Zugangsnetzwerke

Zur Erlangung des akademischen Grades eines

**DOKTOR-INGENIEURS (Dr.-Ing.)**

an der Fakultät für

Elektrotechnik und Informationstechnik  
am Karlsruher Institut für Technologie (KIT)

genehmigte

**DISSERTATION**

von

**Dipl.-Ing. Michael Dreschmann**

geb. in Koblenz

Tag der mündlichen Prüfung:

27.11.2015

Hauptreferent: Prof. Dr.-Ing. Dr. h. c. Jürgen Becker

Korreferent: Prof. Dr. Juerg Leuthold

**Realisierung einer adaptiven parallelen Hardwarearchitektur  
für hochperformante OFDMA Zugangsnetzwerke**

1. Auflage: Dezember 2016

©2015 Michael Dreschmann

# Zusammenfassung

Die vorliegende Arbeit beschreibt ein OFDMA basiertes Kommunikationssystem für Zugangsnetzwerke. Ein wichtiges Ziel der vorgestellten Implementierung ist die Erzielung einer hohen Kommunikationsbandbreite von mehr als 50 GBit/s. Gleichzeitig wird das präsentierte System jedoch auch umfassend betrachtet und realisiert, so dass echte Kommunikationsdaten in Form von Ethernetpaketen das Netzwerk passieren können. Mit diesem Ansatz werden sämtliche relevanten Systemkomponenten in einer praktischen und realitätsnahen Umgebung betrachtet und in allen Bereichen des Übertragungssystems Lösungen für Probleme entwickelt, die aufgrund der hohen Datenrate auftreten.

Das entworfene System wurde auf Basis moderner FPGA Hardware realisiert und getestet. Auch wenn viele Systemparameter zur Erlangung der hohen Kommunikationsbandbreite in der konkreten Realisierung statisch festgelegt werden mussten, wurde die zur Implementierung verwendete VHDL Beschreibung des Systems möglichst frei und flexibel konfigurierbar gestaltet. So können aus den entwickelten Funktionsmodulen verhältnismäßig einfache Systeme mit geänderter Parameterkonfiguration erstellt und synthetisiert werden.

Aufgrund der Realisierung eines umfassenden Kommunikationssystems, welches anstelle von Testsignalen die Übermittlung realer Kommunikationsinhalte erlaubt, werden zunächst benötigte technologische Grundlagen aus den unterschiedlichen Bereichen erläutert. Anschließend folgt eine kurze Analyse des aktuellen Standes weiterer Forschungsarbeiten im Bereich OFDM bzw. OFDMA basierter Hochgeschwindigkeitskommunikationssysteme. Es zeigt sich, dass vor allem im Bereich der optischen Übertragungssysteme aktuell an einer Steigerung der Übertragungsrate unter Zuhilfenahme spektral effizienter Verfahren gearbeitet wird.

---

Aus diesem Grund wird auch für diese Arbeit ein konkretes Anwendungsszenario aus dem Bereich der optischen Zugangsnetzwerke gewählt und das entwickelte System auf dessen Anforderungen ausgelegt. Unter Berücksichtigung der technologischen Limitierungen aktueller Hardwarekomponenten, werden im Folgenden die grundlegenden Systemparameter festgelegt. Anschließend werden mittels *Top-Down Ansatz* sämtliche Kommunikationsparameter und Systemkomponenten im Detail spezifiziert. Parallel dazu erfolgt eine Auswahl geeigneter Hardwarekomponenten und es wird stets überprüft, ob die gewählten Realisierungsansätze mit den ausgewählten Komponenten umsetzbar sind.

Im nächsten Schritt folgt schließlich die Implementierung des Kommunikationssystems, wobei aus Performanzgründen nahezu alle Signalverarbeitungsschritte und die Paketdatenverarbeitung der Ethernetkommunikation in Hardware innerhalb mehrerer FPGAs realisiert werden. Die entwickelten Module werden, soweit sinnvoll möglich, mit Hilfe von Konfigurationsparametern flexibel gestaltet, so dass eine Wiederverwendung einzelner Module im Kommunikationssystem einfach möglich ist. Weiterhin ermöglicht es diese Flexibilität das System relativ leicht an geänderte Kommunikationsparameter oder Hardwareplattformen anzupassen.

Neben der Umsetzung der eigentlichen Signalverarbeitung erforderte die Realisierung dieser Arbeit zusätzlich den Aufbau der Systemhardware. Dazu gehören neben entsprechend leistungsfähigen FPGA Boards auch diverse DA und AD Wandler sowie Schnittstellen für (unter anderem) 10G Ethernet. Da zur Erzielung maximaler Performanz mit modernster Hardware gearbeitet wurde, standen für die meisten benötigten Komponenten keine kommerziellen Baugruppen zur Verfügung, so dass diese ebenfalls zu großen Teilen im Rahmen der Arbeit realisiert werden mussten.

Nach der Entwicklung der einzelnen Systemkomponenten wurde das Gesamtsystem integriert und in Betrieb genommen um die korrekte Funktion zu überprüfen. Dabei konnte verifiziert werden, dass sämtliche Teilsysteme korrekt arbeiten, obwohl es leider bis zum Ende der Arbeit nicht möglich war das Gesamtsystem komplett aufzubauen, da nicht alle benötigten FPGA Boards ausreichend schnell hergestellt werden konnten.

---

Abschließend werden die ermittelten Ergebnisse bewertet und eine Übertragbarkeit des erstellten Systems auf andere Kommunikationsumgebungen diskutiert.



# Abstract

The thesis presents an OFDMA based communication system for access networks. An important objective of the presented implementation is the achievement of a high communication bandwidth of more than 50 GBit/s. Simultaneously the presented system is comprehensively examined and realized, allowing real communication data in form of Ethernet packets to pass the network. Following this approach, all relevant system components get considered in a practical and realistic environment and solutions are developed for all parts of the transmission system where problems arise due to high data rates.

The developed system was realized and tested based on modern FPGA hardware. Even though many system parameters need to be fixed to achieve the high communication bandwidth, the VHDL description of the system was created as free and flexible as possible. Thus further systems with other parameter configuration may be created and synthesized comparatively simple.

Due to a realization of an extensive communication system transferring real communication data instead of test signals, technological basics of different domains are explained at the beginning of the thesis. Following a short analysis of related work in the domain of OFDM and OFDMA based high speed communication systems is done. It seems that publications on an increment of transfer rates based on spectral efficient modulation techniques currently can be found mostly in the optical domain.

Due to that reason, also the presented thesis focuses on a concrete application from the field of optical access networks and the developed system is designed to meet its requirements. Considering technological limitations of actual hardware components, basic system parameters are defined. Following the complete set of communication parameters and system components are specified based on a *Top-Down approach*. Simultaneously

---

suitable hardware components are selected while it is permanently verified that the chosen realization approach fits to the selected components.

In the next step the implementation of the communication system follows. Nearly all signal processing parts and packet data processing for the Ethernet communication is realized in hardware with several FPGAs. Whenever possible, the developed modules are designed in a very flexible way using parametrization. So reuse of individual modules within the communication system is easily possible and an additional flexibility is provided allowing adapting the system easily to slightly changed communication parameters or hardware platforms.

Beneath the realization of the actual signal processing, this thesis additionally required the setup of the system hardware. This includes besides powerful FPGA boards also diverse DA and AD converter as well as several interfaces, e.g. for 10G Ethernet. No commercial development hardware was available due to the fact that state-of-the-art hardware had to be used to reach maximum performance. So also great parts of the system hardware had to be realized as part of the thesis.

After the development of individual components, the overall system was integrated and put into operation to verify its correct functionality. It could be verified that all subsystems operate correctly even though it was not possible to complete the integration of the full system until the end due to some unavailable FPGA boards.

Finally the found results are evaluated and their transferability to other communication environments is discussed.



# Vorwort

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Technik der Informationsverarbeitung (ITIV) des Karlsruher Institut für Technologie (KIT). Während dieser Zeit hatte ich die Gelegenheit wundervolle Menschen kennenzulernen, Einblick in viele spannende Forschungsthemen zu erhalten und auch ein wenig die Welt zu erkunden. Ich denke gerne zurück an diese schöne Zeit und möchte mich hiermit bei allen bedanken, die mich auf diesem Weg begleitet haben.

Ein ganz besonderer Dank geht an meinen Doktorvater Prof. Jürgen Becker, der mich bereits früh im Studium unterstützte und die Promotion am ITIV ermöglichte. Seine exzellente Betreuung und sein großes Vertrauen in mich und meine Arbeit erlaubten es mir in meiner Zeit am ITIV stets mit viel Freude eigene Ideen zu verwirklichen. Ebenfalls herzlich bedanken möchte ich mich bei Prof. Juerg Leuthold, nicht nur für die Übernahme des Koreferats, sondern auch für eine hervorragende Unterstützung während meiner Promotion.

Auch an Prof. Michael Hübner von der Ruhr-Universität Bochum geht ein herzlicher Dank für das in mich gesetzte Vertrauen und die Möglichkeit bereits früh im Studium in die wissenschaftliche Arbeit am ITIV einzusteigen.

Ebenfalls danken möchte ich meinen einzigartigen Kollegen, die mich zum Teil über ein Jahrzehnt am ITIV begleitet haben und jeden einzelnen Tag besonders machten. Besonders erwähnen möchte ich hier Joachim Meyer, der mich als treuer Kollege und Freund in allen Situationen begleitet hat. Auch Oliver Sander als langjähriger Begleiter und Freund stand mir stets mit Rat und Tat zur Seite. Ein weiter Dank geht aber auch an Falco Bapp, Lukas Meder, Stefanie Friederich, Alexander Klimm, Michael Rückauer und all die anderen Kollegen am ITIV, deren herausragende

---

Unterstützung und offene Art viel zum Erfolg dieser Arbeit beigetragen haben. Ohne euch wäre die Zeit am ITIV nicht die selbe gewesen. Auch René Schmogrow und Phillip Schindler vom IPQ gebührt ein herzlicher Dank für aufschlussreiche Diskussionen und für ihre Unterstützung im Bereich der Optik und Hochfrequenztechnik, ohne die diese Arbeit in ihrer jetzigen Form nicht möglich gewesen wäre.

Weiterhin möchte ich mich noch bei allen bedanken, die durch ihre Anregungen, Korrekturen und Diskussionen auf die eine oder andere Weise zur Erstellung dieser Arbeit beigetragen haben. Besonders erwähnen möchte ich hier Martina Gassner für ihre Unterstützung beim Korrekturlesen der Arbeit.

Schließlich geht ein ganz spezieller Dank auch an meine Familie, ganz besonders an meine Eltern Wolfgang und Gisela als auch an Sascha und Nicole. Auch sie unterstützten mich stets auf dem Weg zu dieser Dissertation.

Karlsruhe, im November 2015  
Michael Dreschmann

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	3
1.2. Zielsetzung und Abgrenzung der Arbeit . . . . .	5
1.3. Aufbau der Arbeit . . . . .	6
<b>2. Grundlagen</b>	<b>9</b>
2.1. Digitale Kommunikation . . . . .	9
2.1.1. Basisbandkommunikation . . . . .	9
2.1.2. Trägermodulation . . . . .	10
2.1.3. Komplexe Basisbanddarstellung . . . . .	14
2.2. Mehrträger Modulationsverfahren . . . . .	18
2.2.1. OFDM . . . . .	20
2.2.2. Fast Fourier Transformation . . . . .	23
2.2.3. Autokorrelation basierte Synchronisation . . . . .	26
2.2.4. Pilottöne . . . . .	29
2.3. Spezialisierte Funktionseinheiten . . . . .	29
2.3.1. Numerically Controlled Oscillator . . . . .	30
2.3.2. Direct Digital Synthesis . . . . .	31
2.3.3. Phase Locked Loop . . . . .	32
2.3.4. CORDIC Algorithmus . . . . .	33
2.3.5. PID-Regler . . . . .	36
2.4. Field Programmable Gate Array . . . . .	37
2.4.1. Aufbau . . . . .	37
2.4.2. Multi-Gigabit Transceiver . . . . .	39

<b>3. Stand der Technik</b>	<b>43</b>
3.1. OFDM in optischen Kommunikationssystemen . . . . .	44
3.2. Rechenarchitekturen für OFDM basierte Kommunikati- onssysteme . . . . .	46
3.3. Funktionseinheiten . . . . .	47
3.3.1. FFT Berechnung . . . . .	47
3.3.2. Zeit- und Frequenzsynchronisation . . . . .	49
3.3.3. Fehlerkorrektur . . . . .	50
<b>4. Systemarchitektur</b>	<b>53</b>
4.1. Anforderungen . . . . .	54
4.1.1. Einsatzbereich des Kommunikationssystems . . . .	54
4.1.2. Analog-Elektronisches und Optisches System . . .	56
4.1.3. Eingesetzte Hardware . . . . .	59
4.1.4. Aufgabenverteilung und Netzsynchronisation . . .	64
4.1.5. Bandbreitenverteilung . . . . .	66
4.2. Spezifikation der Systemparameter und Kommunikations- protokolle . . . . .	67
4.2.1. Grundlegende Systemparameter . . . . .	67
4.2.2. OFDM Signalaufbau im Frequenzbereich . . . . .	68
4.2.3. OFDM Signalaufbau im Zeitbereich . . . . .	71
4.2.4. Synchronisation . . . . .	75
4.2.5. Einbettung von Kontroll- und Nutzdaten . . . . .	78
4.3. Hardware Architektur . . . . .	84
4.3.1. Downlink . . . . .	84
4.3.2. Uplink . . . . .	89
4.3.3. Nutzdatentransfer und dynamische Bandbreitenal- lokation . . . . .	92
4.4. Gesamtsystemübersicht . . . . .	99
<b>5. Realisierung der Signalverarbeitung</b>	<b>101</b>
5.1. Designkriterien für das FPGA System . . . . .	102

5.2.	Fast Fourier Transformation . . . . .	105
5.2.1.	Radix Butterfly . . . . .	107
5.2.2.	Drehfaktormultiplikationen . . . . .	109
5.2.3.	Aufbau der FFT . . . . .	112
5.3.	Zeit- und Frequenzsynchronisation . . . . .	114
5.3.1.	Autokorrelation . . . . .	116
5.3.2.	Spitzendetektion . . . . .	117
5.3.3.	CORDIC Algorithmus . . . . .	118
5.3.4.	Numerically controlled oscillator . . . . .	119
5.3.5.	Equalizer . . . . .	119
5.3.6.	Frequenzoffsettracking . . . . .	120
5.3.7.	Synchronisation des Uplinks . . . . .	121
5.4.	Modulatoren und Demulatoren . . . . .	122
5.5.	Paketdatenverarbeitung . . . . .	125
5.6.	I/O Schnittstellendesign . . . . .	126
5.6.1.	ADC/DAC Anbindung an OLT . . . . .	126
5.6.2.	ADC/DAC Anbindung an ONU . . . . .	130
5.6.3.	Inter-Board Kommunikation . . . . .	132
<b>6.</b>	<b>Bereitstellung der Hardwareplattform</b>	<b>139</b>
6.1.	FPGA Mainboard . . . . .	140
6.2.	Stromversorgung . . . . .	144
6.3.	DA und AD Wandler für die Teilnehmeranschlüsse . . . . .	147
6.4.	DDS Taktgenerator . . . . .	150
6.5.	Analog Frontends . . . . .	153
<b>7.</b>	<b>Testergebnisse</b>	<b>159</b>
7.1.	Komponententests . . . . .	159
7.1.1.	FFT Tests . . . . .	160
7.1.2.	IO Interfaces . . . . .	164
7.2.	Systemsimulation . . . . .	167

## Inhaltsverzeichnis

---

7.3. Systemtests . . . . .	171
7.3.1. Packetdatenverarbeitung . . . . .	172
7.3.2. OFDM Signalverarbeitung . . . . .	173
<b>8. Schlussfolgerung und Ausblick</b>	<b>177</b>
8.1. Zusammenfassung und Bewertung der Ergebnisse . . . . .	177
8.2. Übertragbarkeit auf andere Kommunikationssysteme . . . . .	178
8.3. Ausblick . . . . .	179
<b>A. Anhang</b>	<b>183</b>
Abbildungsverzeichnis . . . . .	183
Tabellenverzeichnis . . . . .	187
Abkürzungsverzeichnis . . . . .	191
<b>Literatur- und Quellennachweise</b>	<b>195</b>
<b>Betreute studentische Arbeiten</b>	<b>201</b>
<b>Eigene Veröffentlichungen</b>	<b>203</b>

# 1. Einleitung

1888, vor nicht einmal 150 Jahren, gelang es Heinrich Hertz hier in Karlsruhe die von James Clerk Maxwell vorhergesagten elektromagnetischen Wellen experimentell nachzuweisen. Mit dieser Entdeckung wurde der Grundstein für eine Revolution in der Kommunikationstechnik gelegt, die bis heute anhält und die Gesellschaft nachhaltig geprägt hat. Das in der Folge entwickelte Verständnis der elektromagnetischen Wellen und dem mathematischen Verhalten von Signalen ist dabei nicht nur grundlegend für die Realisierung jeder Art von funktechnischen Anlagen. Auch für leitungsgebundene und optische Kommunikationskanäle wird ihr Verhalten im Zuge moderner Kommunikationsverfahren zunehmend wichtig.

Schnell nach ihrer Entdeckung wurde der Nutzen elektromagnetischer Wellen für die Kommunikationstechnologie erkannt und bereits 1901 war es möglich, mit Hilfe einer Funkverbindung über den Atlantik zu kommunizieren. Anfänglich beschränkte man sich dabei auf die Telegraphie, aber bereits 1913 gelang Alexander Meißner die Erzeugung ungedämpfter elektrischer Schwingungen, wodurch die Übermittlung von komplexeren Signalen wie Sprache, Bildern oder digitalen Daten erst möglich wurde. 10 Jahre später, 1923, wurde der Rundfunk in Deutschland eingeführt und spätestens ab diesem Zeitpunkt zog das Kommunikationsmedium Funk auch immer größere Gesellschaftsgruppen in seinen Bann.

Über lange Zeit erfolgte die Übertragung von Sprach- sowie Bildsignalen ausschließlich analog. 1935 begannen die ersten analogen Fernsehübertragungen und 1958 wurde mit dem A-Netz das erste analoge Mobilfunksystem in Deutschland gestartet. Es dauerte noch eine ganze Weile, bis die Computertechnik so leistungsfähig und vor allem kostengünstig wurde, dass sich die direkte Übertragung von digitalen Signalen mit Hilfe elektromagnetischer Wellen durchzusetzen begann. In der heutigen Zeit ist es kaum noch vorstellbar, dass noch vor 25 Jahren praktisch sämtliche

## 1. Einleitung

---

Massenkommunikation analog erfolgte. Der Beginn der Umstellung des Telefonnetzes auf eine digitale Betriebsart erfolgte 1989 mit dem Start von ISDN und drei Jahre später startete mit dem D-Netz auch das erste digitale Mobilfunknetz auf GSM Basis in Deutschland. Einige Jahre später begann schließlich auch die Digitalisierung der bis dahin noch analog ausgestrahlten Radio- und Fernsehprogramme, so dass heute praktisch alle Massenkommunikationsmedien digital zur Verfügung stehen.

Ein Hauptgrund für die Digitalisierung der Netze war sicherlich eine höhere und gleichbleibende Qualität der übertragenen Signale. Weiterhin lässt sich besonders bei modernen Audio- und Videokompressionsverfahren beobachten, dass die benötigte spektrale Bandbreite für eine Übertragung in digitalisierter Form im Vergleich zu einer analogen Übermittlung in ähnlicher Qualität deutlich kleiner ist. Auch die Möglichkeit, die Übertragungen mittels leistungsfähiger Verschlüsselungsalgorithmen vor dem Zugriff durch Unbefugte zu sichern, ist ein wertvoller Vorteil einer digitalen Signalübertragung.

Zeitgleich mit der Digitalisierung bestehender Kommunikationssysteme zeichnete sich das Aufkommen eines völlig neuen, rein digitalen Kommunikationsmediums ab. Das Internet ermöglichte, im Gegensatz zum traditionellen Rundfunk, erstmals eine bidirektionale und individuelle Kommunikation und damit viele interessante und völlig neue Möglichkeiten. Innerhalb der letzten 20 Jahre hat es sich von einem Nischendasein mit relativ kleinem Nutzerkreis zum primären Kommunikationsmittel entwickelt und beginnt heute bereits mit Diensten wie z.B. Voice-Over-IP und IPTV die traditionellen Kommunikationsmedien abzulösen. Auch die sozialen Auswirkungen eines individuellen und bidirektionalen Kommunikationsmediums auf die Gesellschaft sind nicht zu unterschätzen. Es war in der Menschheitsgeschichte niemals zuvor so einfach an Informationen zu gelangen und immer häufiger werden Missstände auf der Welt nicht durch klassische Recherchen der großen Medienanstalten, sondern mit Hilfe des Internets aufgedeckt (vgl. z.B. die NSA Enthüllungen im Sommer 2013 und die Berichte über den arabischen Frühling im Jahr 2011). In Zukunft sind mit weiter steigenden Datenraten auch noch viele weitere Anwendungsfälle denkbar, z.B. im Bereich der Virtual-Reality.



### 1.1. Motivation

In jüngster Zeit zeichnet sich ein Trend weg vom stationären hin zum mobilen Internetzugang ab, der immer und überall verfügbar ist. So wird eine global vernetzte Gesellschaft auf Basis von Smartphones und ähnlichen Geräten erst möglich und entsprechend rasant steigen aktuell die benötigten Übertragungsraten in Mobilfunknetzen.

Das rasante Wachstum der digitalen Kommunikationsmedien stellt die Technik jedoch auch vor immer größere Herausforderungen. Aufgrund der typischerweise bidirektionalen und individuellen Kommunikationsnatur sind Rundfunkausstrahlungen wie sie seit langer Zeit bei Radio und TV zum Einsatz kommen nicht anwendbar. Vielmehr benötigt jeder Teilnehmer einen dedizierten, bidirektionalen Kommunikationskanal, so dass mit zunehmender Teilnehmerzahl auch der Bandbreitenbedarf in die Höhe schnellt. Zudem werden viele Anwendungen aufgrund von immer häufiger eingesetzten und höher aufgelösten Audio- und Videoübertragungen auch immer bandbreitenhungriger. Alles in allem steigt die benötigte Kommunikationsbandbreite exponentiell an und bisher ist kein Ende in Sicht. Im Gegensatz dazu ist die verfügbare Bandbreite, vor allem bei Funkanwendungen, aber auch zunehmend bei kabelgebundener und optischer Kommunikation begrenzt. Durch eine Erhöhung der zur Signalübermittlung genutzten Trägerfrequenzen lassen sich zwar neue Bandbreitenbereiche erschließen, die Signalausbreitungseigenschaften und die Kosteneffizienz verschlechtern sich bei höheren Frequenzen jedoch typischerweise deutlich. Entsprechend fokussiert sich bereits seit einiger Zeit ein großer Teil der Forschung im Bereich der Kommunikationstechnologien auf die Verbesserung der Ausnutzung verfügbarer Bandbreiten.

Wurden zu Beginn des digitalen Kommunikationszeitalters noch verhältnismäßig einfache Modulationsarten wie z.B. eine Basisbandsignalisierung bei ISDN oder GMSK bei GSM eingesetzt, kommen heute aufgrund ihrer besseren spektralen Effizienz und Anpassbarkeit an den Übertragungskanal vornehmlich Mehrträgerverfahren mit QAM modulierten Einzelträgern zum Einsatz. Ein typischer Vertreter dieser Verfahren ist OFDM, welches z.B. bei aktuellen Mobilfunk-, digital TV und WLAN Standards eingesetzt wird. Da diese Verfahren bereits sehr nahe am Shannon Limit arbeiten (vgl. [38]), setzen neuere Methoden zur weiteren

## 1. Einleitung

---

Nutzbandbreitensteigerung zunehmend auf das Multiplexen mehrerer räumlich getrennter Signalströme mit Hilfe mehrerer Antennen. Vertreter dieser Technologien werden unter dem Namen Multiple Input Multiple Output (MIMO) ([16]) im Funkbereich oder Vectoring im kabelgebundenen Bereich zusammengefasst.

Mit Hilfe dieser neuen Modulationsverfahren sind erstaunliche Nutzbandbreitenzuwächse bei gleichbleibender Belegung im verfügbaren elektromagnetischen Spektrum möglich. Häufig wird dabei jedoch verdrängt, welchen enormen Aufwand solche komplexe Übertragungsverfahren an die Sende- und vor allem Empfangshardware stellen. Abgesehen von der Verschiebung des Kommunikationssignals zwischen dem Basisband und einer Trägerfrequenz, findet die Signalprozessierung heute praktisch vollständig digital statt. Der Aus- bzw. Eingang der entsprechenden Mischer-Stufen wird typischerweise direkt mit Hilfe von AD bzw. DA Wandlern in eine digitale Verarbeitungsdomäne überführt. Das bedeutet, dass praktisch sämtliche Parameter des Übertragungssystems und auch dessen Leistungsfähigkeit primär durch die Qualität der eingesetzten digitalen Signalverarbeitung bestimmt wird. War der Aufwand beim Einsatz von Einträgermodulationsverfahren noch verhältnismäßig gering, so erfordern Mehrträgerverfahren bereits die simultane Verarbeitung vieler Trägersignale in der digitalen Signalverarbeitungseinheit. Mit dem Einzug der MIMO Technologie ergeben sich noch einmal deutlich höhere Anforderungen an die Rechenleistung der eingesetzten Hardware. Die in der Praxis beschränkten Möglichkeiten sind auch einer der Gründe, weshalb bisher maximal 4x4 MIMO Systeme zum Einsatz kommen. Weitere, z.T. erhebliche Anforderungen an die Rechenleistung der digitalen Hardware ergeben sich aus zusätzlichen Anforderungen des Kommunikationsnetzes. Beispielsweise erfordert eine Punkt-zu-Mehrpunktverbindung, wie sie typischerweise in Mobilfunk- oder optischen Zugangsnetzen zum Einsatz kommt, einen gewissen Aufwand, um alle Kommunikationsteilnehmer zu synchronisieren und diesen Zustand aufrecht zu halten.

Es ist in allen Bereichen der Kommunikationstechnik ein Trend zur Digitalisierung der Informationsübertragung zu erkennen und in vielen Systemen kommen dabei mittlerweile äußerst komplexe und damit sehr rechenintensive Übertragungsformen zum Einsatz. Zuletzt hat sich dieser Trend auch im Bereich der optischen Kommunikationstechnik auf Basis von Glasfasern bemerkbar gemacht. Dieses Medium bietet physikalisch

bedingt bereits sehr große Bandbreitenreserven, so dass hier bisher noch nicht der Einsatz extrem komplexer Modulationstechniken notwendig ist. Vor allem im Bereich der Zugangsnetze stößt allerdings auch die Glasfaser immer mehr an ihre Grenzen, so dass auch hier ein steigendes Interesse an spektral effizienten Modulationstechniken besteht. Als zusätzliche Schwierigkeit kommt in diesem Bereich hinzu, dass die typischerweise genutzten Bandbreiten um Größenordnungen über denen in funk- und kupferbasierten Technologien liegen. Entsprechend performant muss daher auch die eingesetzte digitale Signalverarbeitung arbeiten.

Bereits jetzt werden viele rechenintensive Bestandteile der Signalverarbeitung von Kommunikationssystemen wie z.B. die Fast Fourier Transformation (FFT) mit Hilfe dedizierter Hardwarebeschleuniger verarbeitet. In Zukunft werden diese Hardwarebeschleuniger z.T. noch deutlich erweitert werden müssen, damit in Systemen mit sehr hohen Bandbreitenanforderungen beispielsweise gleich mehrere Samples pro Takt verarbeitet werden können.

## 1.2. Zielsetzung und Abgrenzung der Arbeit

Anknüpfend an die zu beobachtende stetige Steigerung der benötigten Performanz im Bereich der digitalen Signalverarbeitung in Kommunikationssystemen soll in dieser Arbeit untersucht werden, welche Möglichkeiten dazu im Einzelnen für ein praktisch realisierbares System zur Verfügung stehen und welche Performanz mit verhältnismäßig kostengünstigen Mitteln heute bereits realisierbar ist. Besonderen Wert wurde dabei auf eine umfassende Betrachtung des Signalverarbeitungssystems gelegt. Dazu gehören insbesondere auch alle Schnittstellen zu Wandlern zwischen der analogen und digitalen Welt sowie die Verarbeitung von realen Daten in Form von Ethernet Paketen, so dass am Ende ein vollständig funktionierendes Kommunikationssystem entsteht, welches zur transparenten Übertragung von Netzwerkverkehr genutzt werden kann. Da eine Übertragung bei der angepeilten hohen verwendeten Bandbreite lediglich auf einem optischen Kanal sinnvoll anwendbar ist, wird das System für ein glasfaserbasiertes, optisches Zugangsnetzwerk entworfen. Ein wesentlicher Optimierungspunkt ist die maximale Performanz und

entsprechend wird die Auslegung des Systems auch an vielen Stellen davon getrieben. Die Erstellung eines möglichst flexiblen Systems ist ausdrücklich nicht Ziel dieser Arbeit. Entsprechende Implementierungen sind beispielsweise bereits in [24] zu finden.

### 1.3. Aufbau der Arbeit

Der Aufbau der Arbeit gestaltet sich wie folgt:

Zunächst werden in Kapitel 2 einige Grundlagen zum Verständnis der in den folgenden Kapiteln behandelten Themenbereiche erläutert. Da diese Arbeit ein breites Spektrum von Methoden und Verfahren aus teilweise recht unterschiedlichen technischen Bereichen heranzieht, würde eine ausführliche und umfassende Behandlung jedes Aspekts den Rahmen dieses Kapitels sprengen. Es wurde jedoch versucht, die grundlegenden Funktionsweisen aller in dieser Arbeit eingesetzten Verfahren hinreichend klar darzustellen, so dass ein Verständnis späterer Kapitel problemlos möglich ist. Zunächst wird die grundsätzliche Funktionsweise digitaler Kommunikation erläutert und anschließend etwas genauer auf mehrträgerbasierte Modulationsverfahren eingegangen, da diese Arbeit wesentlich darauf basiert. Anschließend werden einige elementare Funktionseinheiten präsentiert, welche an verschiedenen Stellen der späteren Implementierung immer wieder auftauchen. Zum Abschluss erfolgt schließlich noch eine kurze Übersicht zur grundsätzlichen Funktionsweise von Field Programmable Gate Arrays (FPGAs), die für diese Arbeit genutzte Basistechnologie, mit einem besonderen Augenmerk auf Multi-Gigabit-Transceiver (MGTs), welche sehr häufig zur Anbindung hochperformanter Peripherie zum Einsatz kommen.

Im nächsten Kapitel wird der Bezug des hier vorgestellten Systems zu weiteren Arbeiten aus dem Bereich der Mehrträger Kommunikationsverfahren, optischen Netzwerken sowie Hochgeschwindigkeitsdatenverarbeitung hergestellt und die hier vorgestellte Arbeit wird entsprechend abgegrenzt.

Die Kapitel 4, 5 und 6 bilden den Kern dieser Arbeit. In ihnen wird der Reihe nach die konkrete Funktionsweise des Kommunikationssystems sowie dessen Realisierung auf FPGA- sowie auf Hardwareebene beschrie-

ben. Dabei werden in Kapitel 4 der genaue Aufbau des übermittelten Datensignals in Zeit- und Frequenzbereich sowie die dazu benötigten Signalverarbeitungskomponenten beschrieben. Neben der Architektur des Systems werden immer auch die Gründe, welche zu gewissen Entscheidungen führten, genannt. Kapitel 5 behandelt anschließend die genaue Realisierung ausgewählter Funktionsblöcke des Kommunikationssystem auf der Zielhardware, welche wegen der hohen Kommunikationsbandbreite besonders aufwändig bzw. komplex aufgebaut sind. Aufgrund des großen Umfangs des Signalverarbeitungssystems werden einfache und trivial realisierbare Funktionseinheiten nicht im Detail erläutert, wenn ihr Aufbau etwa aufgrund vernachlässigbarem Ressourcenverbrauch unkritisch ist. Da zu dieser Arbeit auch die Realisierung der Systemhardware in Form von entsprechenden Platinen und elektronischen Schaltungen zählt, wird deren Aufbau in Kapitel 6 behandelt.

Kapitel 7 fasst die ermittelten Ergebnisse der Arbeit zusammen. Dabei handelt es sich nur zum Teil um klassische Messergebnisse, da wertvolle Resultate vor allem auch im Bereich der Systemarchitektur in Kapitel 4 sowie deren Realisierung in Kapitel 5 und 6 zu finden sind.

Im letzten Kapitel 8 werden die ermittelten Ergebnisse bewertet und ihre Übertragbarkeit auf andere Formen von Kommunikationssystemen diskutiert sowie entsprechende Grenzen aufgezeigt.



## 2. Grundlagen

In diesem Kapitel werden wichtige Grundlagen diskutiert, welche zum Verständnis der nachfolgenden Beschreibung des Orthogonal-Frequency-Division-Multiple-Access (OFDMA) Kommunikationssystems benötigt werden. Es werden zunächst die grundsätzlichen Prinzipien digitaler Informationsübertragung erläutert sowie später auf konkrete Funktionsblöcke der hier betrachteten Orthogonal-Frequency-Division-Multiplexing (OFDM) Kommunikation eingegangen. Weiterhin erfolgt am Ende des Kapitels eine grobe Übersicht über wesentliche Eigenschaften der zur Realisierung des Systems verwendeten Virtex-6 FPGA Technologie.

### 2.1. Digitale Kommunikation

#### 2.1.1. Basisbandkommunikation

Es ist eine grundlegende Eigenschaft digitaler Information, dass diese in Form von diskreten Werten vorliegt. Ein digitaler Datenstrom, welcher mit Hilfe eines digitalen Kommunikationssystems übertragen werden kann, besteht aus einer Folge diskreter Werte. Man spricht von einem seriellen digitalen Datenstrom, wenn die Werte der Folge lediglich 0 oder 1, d.h. die elementare Einheit der Information, ein Bit, einnehmen können. Ein paralleler digitaler Datenstrom liegt dann vor, wenn sich die einzelnen Werte der Folge aus mehreren Bits zusammensetzen und somit parallel in einem Schritt übermittelt werden. Ein typisches Beispiel für einen parallelen Datenstrom ist ein Datenbus eines Prozessorsystems, welcher in jedem Schritt (Takt) zwischen 8 und 64 Bit (je nach Leistungsfähigkeit des Prozessors) an Information übermittelt.

Bei einfachen digitalen Kommunikationsschnittstellen, wie sie beispielsweise innerhalb von Integrated Circuits (ICs) oder auf Leiterplatten zwischen verschiedenen ICs eingesetzt werden, erfolgt die Übermittlung der diskreten Werte direkt mit Hilfe von unterschiedlichen Spannungspegeln über eine elektrische Leiterbahn. Ein zusätzliches Taktsignal gibt dabei häufig die Übertragungsschritte der Folge diskreter Werte vor (Beispiel *SPI* [43]). Diese Übermittlung kann als eine Art Amplitudenmodulation im Basisband betrachtet werden. Der Begriff *Basisband* bedeutet in diesem Zusammenhang, dass für die Übermittlung des digitalen Signals keinerlei Zwischenfrequenzen eingesetzt werden und sich die Bandbreite der Übertragung bis hinunter zu 0 Hz erstreckt.

Ein ähnliches Vorgehen wurde auch bei den ersten Netzwerken für Computer eingesetzt, beispielsweise bei 10BASE-T Ethernet. Das *BASE* im Namen steht dabei für *Basisbandübertragung*. Allerdings wurde durch die Verwendung eines entsprechenden Linecodes die Übermittlung eines dedizierten Taktsignals eingespart. Der eingesetzte Manchestercode erlaubt die direkte Rückgewinnung der einzelnen Folgeschritte aus dem seriellen Datenstrom.

Eine rein amplitudenmodulierende Basisbandübertragung lässt sich sehr einfach realisieren, weshalb sie auch heute noch die mit Abstand dominierende Übertragungsform für kurze, leitungsgebundene, digitale Übertragungstrecken in ICs und auf Leiterplatten ist. Für die Informationsübertragung auf weiten Strecken ist sie aber denkbar ungeeignet. Durch die einfache, häufig binäre Amplitudenmodulation ist die Bandbreitenausnutzung sehr schlecht und die Übertragung im Basisband erlaubt zudem nur eine Übertragung über relativ kurze Distanzen. Für die digitale Informationsübertragung in Weitverkehrsnetzen (wozu technisch auch das in dieser Arbeit vorgestellte optische Zugangnetzwerk zählt) kommen daher deutlich komplexere Übertragungsverfahren zum Einsatz.

### 2.1.2. Trägermodulation

Grundsätzlich erfolgt eine Informationsübermittlung immer mit Hilfe eines Trägersignals. Auch eine Basisbandübertragung kann so beschrieben werden, wenn man bedenkt, dass die Trägerfrequenz für den Fall *Basisband* zu 0 Hz definiert ist. Für die meisten funktechnischen Anwen-



dungen, aber auch für lange drahtgebundene Kommunikationssysteme (z.B. Digital Subscriber Line (DSL)) ist sie jedoch ungleich Null, da sich Signale in höheren Frequenzbereichen einfacher übertragen lassen. Speziell im funktechnischen Bereich wird durch die unterschiedliche Wahl der Trägerfrequenzen verschiedener Systeme die parallele Nutzung des Frequenzspektrums durch viele Anwendungen auch erst ermöglicht. Für den Fall, dass eine Trägerfrequenz ungleich Null ist, spricht man nicht mehr von einer Basisbandübertragung, sondern von einem modulierten Träger.

Wie die folgende Gleichung veranschaulicht, existieren drei grundlegende Parameter, durch die ein Trägersignal definiert bzw. zur Informationsübermittlung moduliert werden kann:

$$T(t) = A(t) * \cos(2 * \pi * (f(t) + f_c) * t + \phi(t))$$

Dabei handelt es sich um die Amplitude  $A$ , die Frequenz  $f$  sowie die Phase  $\phi$ . Der Parameter  $f_c$  gibt die konstante Trägerfrequenz an. Eine Informationsübermittlung kann daher nicht nur durch eine Variation der Amplitude erfolgen, sondern auch durch die Modulation von Frequenz oder Phase. Während Amplitude und Frequenz absolut messbare Größen sind, ist die Phase lediglich in Relation zu einer Referenzschwingung mit bekannter Phase definiert. Unter anderem aus diesem Grund ist die Auswertung der Phase etwas schwieriger und kam in früheren, analogen Systemen nur selten zum Einsatz. In heutigen, digitalen Systemen wird in regelmäßigen zeitlichen Abständen ein Signal als Phasenreferenz übermittelt, dessen Phasenlage dem Empfangssystem aufgrund der Systemspezifikation bekannt ist und somit als Bezugspunkt für die Dekodierung phasenmodulierter Träger verwendet wird.

Auch die Amplitude  $A$  variiert zum Teil recht stark mit der Empfangsfeldstärke am Empfänger und muss nachjustiert werden. Das erfolgt typischerweise mit Hilfe einer Automatic Gain Control (AGC) Schaltung, welche das Signal so verstärkt, dass die mittlere Amplitude einem vorgegebenen Wert entspricht. Handelt es sich dabei um ein amplitudenmoduliertes Signal, welches sehr unausgewogen moduliert wurde, kann mit einer AGC nicht immer die richtige Verstärkung erreicht werden, da der Mittelwert des Empfangssignals in diesem Fall vom Mittelwert des Aussteuerbereichs der Amplitude stark abweicht. Daher verwenden moderne, digitale Empfänger auch für die Nachführung der Empfän-

## 2. Grundlagen

---

gerverstärkung ein Referenzsignal, welches in regelmäßigen Abständen ausgesendet wird. Lediglich die Frequenz  $f$  lässt sich auch unter stark variierenden Empfangsbedingungen einfach und zuverlässig auswerten. Daher wurden früher für hochwertige Übertragungen häufig frequenzmodulierte Systeme eingesetzt (z.B. UKW Rundfunk).

Als Vorgriff kann hier bereits angemerkt werden, dass in OFDM Systemen sehr viele verschiedene Träger zum Einsatz kommen, deren Amplitude bzw. Phase basierend auf Referenzsignalen nachgeführt werden müssen. Das ist die Aufgabe der sog. Kanal-Equalizer, welche im Empfänger direkt hinter der Berechnung der Fast Fourier Transform (FFT) folgen.

Ein Trägersignal kann somit über eine Variation von Amplitude, Phase oder Frequenz moduliert werden. Diese Modulation erfolgt bei der Verwendung von analogen Signalen kontinuierlich oder bei digitalen Datenströmen diskret, indem je nach zu modulierendem digitalen Wert eine bestimmte Amplitude, Frequenz oder Phase ausgesendet wird. Entsprechend wurden unterschiedliche Bezeichnungen für verschiedene Modulationsverfahren etabliert. *Amplitude-Modulation (AM)*, *Frequency-Modulation (FM)* und *Phase-Modulation (PM)* bezeichnen dabei die analoge Amplituden-, Frequenz- und Phasenmodulation, während deren grundlegende digitale Pendant mit *Amplitude-Shift-Keying (ASK)*, *Frequency-Shift-Keying (FSK)* sowie *Phase-Shift-Keying (PSK)* bezeichnet werden. *Shift Keying* steht dabei für eine Kodierung durch die Verschiebung des entsprechenden Modulationsparameters auf unterschiedliche diskrete Stufen.

Es existieren noch viele weitere Variationen dieser grundlegenden digitalen Modulationsarten. Beispielsweise bezeichnet *Differential-Phase-Shift-Keying (DPSK)* eine differentielle Phasenkodierung. Das bedeutet, die zu übertragende Information wird nicht durch die Phasenlage selbst bestimmt, sondern durch den Wert der Änderung der Phasenlage zwischen zwei aufeinander folgenden Übertragungsschritten.

Neben der Modulation lediglich eines Trägerparameters ist es auch möglich, Amplitude und Frequenz oder Amplitude und Phase simultan zu modulieren. Die gemeinsame Variation von Frequenz und Phase ist nicht sinnvoll, da es sich bei der Frequenz um die Ableitung der Phase handelt und somit eine unabhängige Modulation der beiden Parameter nicht möglich ist. In der Praxis wird sehr häufig die als Quadrature-Amplitude-Modulation (QAM) bezeichnete kombinierte Amplituden- und Phasen-

modulation eingesetzt. D.h. im selben Übertragungsschritt wird gleichzeitig Information mit Hilfe der Amplitude sowie der Phase eines Trägersignals übermittelt. Eine der ersten Anwendungen einer analogen QAM war die Kodierung der Farbinformation in analogen Television (TV) Systemen wie z.B. beim Phase-Alternating-Line (PAL) System. Die Helligkeitsinformation wird dabei wie bereits beim Schwarz-Weiß Fernsehen als amplitudenmoduliertes Signal im Basisband übertragen. Für die Übermittlung der Farbinformation (Farbton bzw. Farbsättigung) werden nun zwei zusätzliche Signale benötigt, welche mit Hilfe einer analogen QAM auf einen Hilfsträger moduliert wurden.

Die meisten modernen digitalen Übertragungssysteme, welche nennenswerte Datenmengen übertragen, verwenden eine diskrete QAM, da sie aufgrund der simultanen Modulation zweier Trägerparameter besonders bandbreiteneffizient ist. Die diskreten Modulationsstufen von Amplitude und Phase werden in der Regel in Form von komplexen Zahlen angegeben und in sog. Konstellationsdiagrammen im komplexen Zahlenraum dargestellt. Abbildung 2.1 zeigt ein 16-QAM Konstellationsdiagramm. D.h. es existieren 16 diskrete Punkte in der zweidimensionalen Ebene, welche sich durch eine entsprechende Wahl von Amplitude und Phase des Trägers darstellen lassen.

Eine konkrete Trägermodulation durch einen Punkt im Konstellationsdiagramm wird auch als Symbol bezeichnet. Die Anzahl an Symbolen, die pro Zeiteinheit übertragen werden, wird mit der sog. Symbolrate angegeben. Da je nach gewähltem Konstellationsdiagramm eine unterschiedliche Anzahl an Konstellationspunkten definiert ist, werden mit jedem übermitteltem Symbol in der Regel mehrere Bits simultan übertragen. Bei einer 16-QAM Konstellation sind das pro Symbol beispielsweise 4 Bit. Die Bandbreite, die vom Übertragungskanal belegt wird, ergibt sich nach Shannon direkt aus der Symbolrate [38].

Die Anzahl der Punkte im Konstellationsdiagramm lässt sich sehr einfach an die Qualität des Übertragungskanals anpassen. Besteht am Empfänger ein hohes Signal-to-Noise-Ratio (SNR), so können mehr Punkte verwendet werden, um gleichzeitig mehr Information zu übertragen. So verwendet Digital Video Broadcasting - Cable (DVB-C) beispielsweise bis zu 256-QAM. Ist die Signalqualität dagegen sehr schlecht oder möchte man sicherstellen, dass wichtige Informationen (z.B. Kontrollkomman-

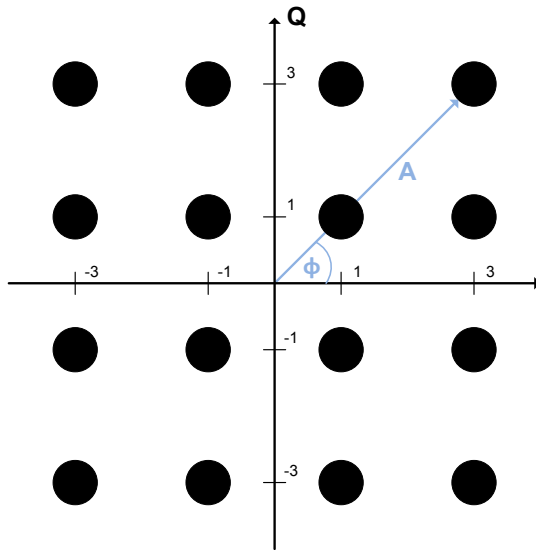


Abbildung 2.1.: QAM-16 Konstellationsdiagramm

dos) sicher und korrekt empfangen werden, verwendet man dagegen z.B. ein Konstellationsdiagramm mit nur zwei Punkten (Binary-Phase-Shift-Keying (BPSK)). Durch die geringere Punkteanzahl vergrößert sich der Abstand zwischen den einzelnen Punkten und ein fehlerfreier Empfang wird trotz schlechterem SNR möglich.

### 2.1.3. Komplexe Basisbanddarstellung

Der direkte Zugriff auf die Phase eines hochfrequenten Trägers ist technisch recht aufwändig. Aus diesem Grund hat sich in modernen Kommunikationssystemen die sog. *in-phase and quadrature component (IQ)*-Basisbandverarbeitung etabliert. Dazu betrachten wir die mathematische Darstellung eines Sendesignals  $S(t)$ :

$$S(t) = A \cos(2\pi f_c t + \phi)$$

Die Frequenz  $f_c$  bildet dabei die Ableitung einer Phase. Der momentane Winkel (oder auch Phase) des Sendesignals ergibt sich durch die Summe des Frequenz- und Phasenterms. Somit lässt sich der momentane Zustand des Sendesignals, welcher auch die Amplitude  $A$  mit einschließt, als komplexe Zahl darstellen und in der komplexen Ebene einzeichnen, siehe Abbildung 2.2.

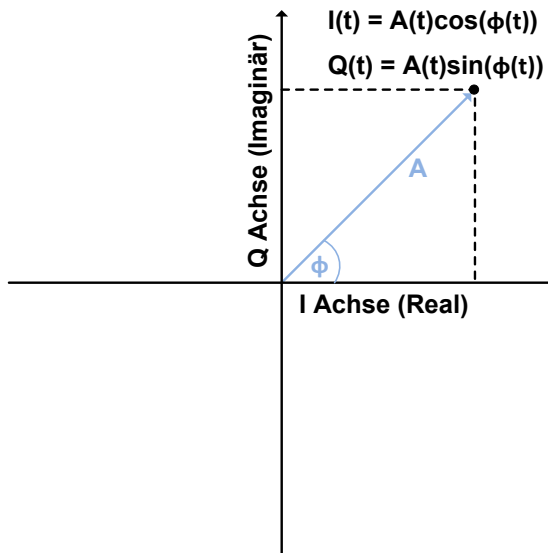


Abbildung 2.2.: Momentanwert einer Sinusschwingung in komplexer Darstellung

In typischen Kommunikationsanwendungen ist der Wert der belegten Bandbreite deutlich geringer als die Trägerfrequenz. D.h. das Sendesignal bewegt sich in einem schmalen Frequenzfenster um die Trägerschwingung, wobei der Träger gleichzeitig als Phasenreferenz dient.

Zur Reduktion der im Sender und Empfänger zu verarbeitenden Signalbandbreite wird in der Regel nicht direkt der momentane Zustand des Sendesignals verarbeitet, sondern die Differenz aus der Trägerschwingung und dem Sendesignal. Dieses Differenzsignal wird auch als IQ-

## 2. Grundlagen

---

Basisbandsignal bezeichnet, wobei der reelle Teil des komplexen Vektors mit  $I$  und der imaginäre Teil mit  $Q$  abgekürzt wird. D.h. der komplexe Vektor, welcher von den beiden Werten  $I$  und  $Q$  aufgestellt wird, beschreibt das Sendesignal in Bezug zu einem Träger, welcher in Amplitude, Frequenz und Phase festgelegt ist. Die Maximalfrequenz des IQ Signals entspricht damit der Maximalbandbreite des Sendesignals.

Wie Eingangs bereits erwähnt, ist die direkte Manipulation der Phase eines Hochfrequenzträgers technisch recht aufwändig. Aus diesem Grund kommen heute sehr häufig sogenannte IQ-Modulatoren bzw. Demodulatoren zum Einsatz, welche das komplexe Basisbandsignal auf einen Träger modulieren bzw. davon extrahieren. Ihre Funktionsweise kann einfach mit Hilfe des Additionstheorems für die trigonometrische Kosinusfunktion erläutert werden:

Das bekannte Additionstheorem

$$\cos(\alpha + \beta) = \cos(\alpha) \cos(\beta) - \sin(\alpha) \sin(\beta)$$

kann leicht zu folgender Form erweitert werden:

$$A \cos(2\pi f_c t + \phi) = A \cos(2\pi f_c t) \cos(\phi) - A \sin(2\pi f_c t) \sin(\phi)$$

Setzt man  $I = A \cos(\phi)$  und  $Q = A \sin(\phi)$  erhält man:

$$A \cos(2\pi f_c t + \phi) = I \cos(2\pi f_c t) - Q \sin(2\pi f_c t)$$

Man erkennt leicht, dass der linke Teil der Gleichung der mathematischen Beschreibung des Sendesignals zu Beginn dieses Abschnitts entspricht.  $I$  und  $Q$  wiederum entsprechen der kartesischen Darstellung des komplexen Basisbandsignals, welches durch die Amplitude  $A$  und den Winkel  $\phi$  beschrieben wird. Zur Erzeugung eines Sendesignals mit definierter Amplitude und Phase, müssen also lediglich Real- ( $I$ ) und Imaginärteil ( $Q$ ) der kartesischen Darstellung des Basisbandsignals mit dem Kosinus bzw. dem Sinus des Trägersignals multipliziert, und das Ergebnis subtrahiert werden. Die Multiplikation entspricht dem elektrischen Mischen von Signalen und ist einfach durchführbar und auch Subtraktionen und Additionen lassen sich bei hohen Frequenzen recht problemlos bewerkstelligen. Die einzige Herausforderung besteht in einer konstanten

Phasenverschiebung des Trägersignals um  $90^\circ$ , da einmal der Sinus und einmal der Kosinus des Trägers benötigt wird. Aber auch diese Aufgabe ist wesentlich einfacher zu bewerkstelligen als eine beliebige Phasenverschiebung des Trägers, wie sie bei einer direkten Modulation durch Amplitude und Phase benötigt würde. Abbildung 2.3 zeigt die technische Realisierung eines IQ Modulators.

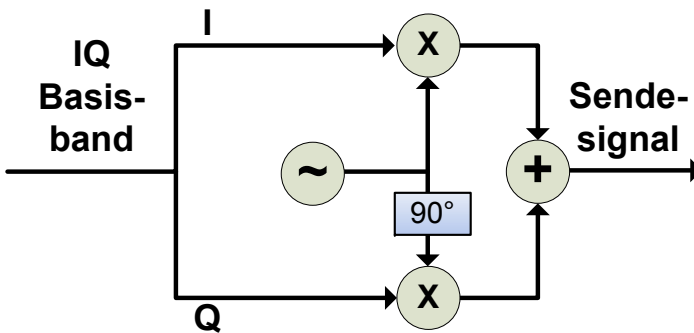


Abbildung 2.3.: IQ Modulator

Da ein IQ Modulator bzw. Demodulator das komplexe Basisbandsignal in kartesischer Form verarbeitet, findet auch die weitere Basisbandsignalverarbeitung in der Regel in dieser Form statt, woher auch der Name der IQ Basisbandverarbeitung rührt. Auch weitere Prozessierungsschritte wie beispielsweise eine FFT arbeiten oft in kartesischer Darstellung am effektivsten. Da die Verarbeitung von IQ Signalen normalerweise hauptsächlich digital stattfindet, soll hier noch auf ein wichtiges Detail hingewiesen werden. Aus dem Nyquisttheorem ist allgemein bekannt, dass die Abtastrate zur digitalen Darstellung eines reellen Signals mindestens dem Doppelten der maximal auftretenden Frequenz entsprechen muss. Bei einem IQ Signal gilt stattdessen, dass die Abtastrate des komplexen Signals mindestens dem Einfachen der maximal auftretenden Frequenz entsprechen muss. Dieser Umstand rührt daher, dass es sich bei einem komplexen Signal eigentlich um zwei reelle Signale handelt, welche gemeinsam entsprechend die doppelte Bandbreite tragen können. Vergleicht man die Modulation einer Trägerschwingung durch ein reel-

les und ein komplexes Basisbandsignal gleicher Frequenz, so stellt man fest, dass sich in beiden Fällen zwei Seitenbänder um den Träger bilden und eine identische Bandbreite belegt wird. Allerdings stellt das untere Seitenband bei der Modulation mit einem reellen Signal eine invertierte Kopie des oberen Seitenbands dar, während sich die beiden Seitenbänder bei einer komplexen IQ Modulation unabhängig mit Information belegen lassen. Die einfache reelle Modulation, wie sie in der Regel in älterer Analogtechnik eingesetzt wird, entspricht einem Sonderfall der IQ Modulation, in der die Q Komponente dauerhaft zu Null gesetzt ist.

Einzelne Frequenzen in den beiden Seitenbändern eines komplex modulierten Trägers lassen sich sehr gut mit Hilfe einer Inverse Fast Fourier Transform (IFFT) kontrollieren, deren komplexer Ausgangsvektor direkt als IQ Basisbandsignal interpretiert wird. Dabei sollte jedoch zwischen logischen und physikalischen Frequenzindizes unterschieden werden. Wird beispielsweise eine 16 Punkt IFFT implementiert, so werden die einzelnen Eingänge, welche zu unterschiedlichen Frequenzen gehören in der Regel von 0 bis 15 durchnummeriert. Dabei handelt es sich um die logische Indizierung. Verwendet man diese Implementierung jedoch zusammen mit einem IQ Modulator so stellt man fest, dass die logischen Indizes 0 bis 7 Frequenzen im oberen Seitenband und die logischen Indizes 8-15 Frequenzen im unteren Seitenband erzeugen. Entsprechend kann für dieselbe IFFT auch eine physikalische Indizierung von -8 bis +7 angegeben werden. Dabei entsprechen die logischen Indizes 0 bis 7 den physikalischen, während von den logischen Indizes 8 bis 15 die Konstante 16 subtrahiert werden muss, um den physikalischen Index zu erhalten. Dieser Zusammenhang wird in Kapitel 4 wieder aufgegriffen, wenn der Einsatz einer IFFT zur Subträgererzeugung in einem OFDM System genauer erläutert wird.

### 2.2. Mehrträger Modulationsverfahren

In den vorherigen Abschnitten wurde die grundsätzliche Funktionsweise digitaler Modulation von Trägersignalen beschrieben. Einige, auch heute noch aktuelle Systeme, verwenden genau diese relativ einfache Art der digitalen Informationsübermittlung. Dazu gehören z.B. DVB-C oder der



weltweit sehr verbreitete Mobilfunkstandard Global System for Mobile Communications (GSM). Bei den steigenden Datenraten moderner Systeme bereitet die einfache, lediglich auf einem Träger basierende Übertragung jedoch vor allem im funktechnischen Bereich zunehmend Probleme. Aufgrund der höheren Datenrate steigt auch die benötigte Symbolrate mit dem der Träger moduliert wird und somit auch dessen belegte Bandbreite. Hieraus ergeben sich zwei Probleme: Zum einen führen *Reflexionen* des Funksignals in urbanen Regionen zur zeitlichen Überlagerung unterschiedlicher Symbole am Empfänger, so dass die Trägerkonstellation nicht mehr so einfach ermittelt werden kann. In älteren Systemen wie dem oben genannten GSM helfen komplexe, dynamisch an den Übertragungskanal angepasste Equalizer das Signal vor der Detektion wieder zu entzerren. Bei steigenden Datenraten werden diese Equalizer jedoch immer komplexer, da sich immer mehr Symbole durch Reflexionen vermischen. Im Fall von DVB-C können aufgrund der leitungsgebundenen Ausstrahlung Reflexionen nicht ohne weiteres entstehen. Zum anderen spricht gegen den Einsatz von Einträgersystemen bei hohen Datenraten die ebenfalls hohe belegte Trägerbandbreite. Aufgrund unterschiedlicher Ausbreitungswege können Teile des von der Übertragung belegten Frequenzspektrums destruktiv interferieren und damit zu einer Auslöschung der Signalenergie dieser Frequenzen führen, wodurch die gesamte Übertragung gestört wird. Dieser Effekt wird auch als *Fading* bezeichnet.

Aus diesem Grund haben sich für moderne digitale Kommunikationssysteme weitgehend Mehrträgerverfahren, auch als Frequency-Division-Multiplexing (FDM) bezeichnet, durchgesetzt. Bei der Verwendung eines Mehrträgerübertragungsverfahrens wird der Datenstrom auf mehrere parallel modulierte Träger, häufig mehrere hundert oder gar tausend, aufgeteilt. Diese werden auch als Subträger bezeichnet, da sie ein Subset eines Datenstroms transportieren. So bleibt die Symbolrate der einzelnen Subträger relativ gering und Signalreflexionen können einfacher toleriert werden. Aufgrund der geringen Symbolrate jedes einzelnen Subträgers belegt dieser auch nur einen kleinen Teil der Gesamtbandbreite der Übertragung. Treten Fadingeffekte oder schmalbandige Störungen auf, so werden lediglich vereinzelte Subträger beeinflusst. Eine in der Regel ohnehin notwendige Fehlerkorrektur bzw. Medienzugriffssteuerung kompensiert den Ausfall einzelner Subträger, so dass aus vereinzelt gestörten Subträgern lediglich eine leichte Abschwächung der Daten-

## 2. Grundlagen

---

rate folgt. Weiterhin besteht auch die Möglichkeit, das zur Modulation verwendete Konstellationsdiagramm für jeden einzelnen Subträger gezielt anzupassen und so bei unterschiedlichen SNRs einzelner Subträger jeweils das passende Modulationsformat zu wählen. Durch die Verwendung vieler Subträger anstelle eines einzelnen Trägers ergibt sich ein sehr robustes Kommunikationssystem, welches auch mit typischen Effekten und Störungen im Funkraum hervorragend klar kommt. Es ist daher nicht verwunderlich, dass praktisch sämtliche modernen digitalen Kommunikationssysteme auf Mehrträger Übertragungsverfahren setzen. Dazu gehören moderne Wireless-Local-Area-Network (WLAN) Standards wie 802.11n oder 802.11ac genauso wie moderne Mobilfunkstandards wie Long Term Evolution (LTE) oder auch terrestrischer Rundfunk nach Digital Video Broadcasting - Terrestrial (DVB-T). Auch die verschiedenen DSL Verfahren setzen auf Mehrträgerkommunikation.

### 2.2.1. OFDM

Eine sehr populäre Variante eines Mehrträgerübertragungsverfahrens ist das sog. OFDM Verfahren. Dabei wird für die einzelnen Subträger der minimal mögliche Abstand im Frequenzspektrum gewählt, der sich aus dem Inversen der für die Subträgermodulation verwendeten Symbolrate ergibt. Abbildung 2.4 zeigt die Energieverteilung der einzelnen Subträger im Frequenzspektrum. Es ist leicht zu erkennen, dass bei einem Energiemaximum eines Subträgers sämtliche benachbarten Subträger ein Energieminimum besitzen. So ist die Packungsdichte der Subträger optimal und die Übertragung des Gesamtdatenstroms mit dem OFDM Verfahren benötigt theoretisch exakt dieselbe Bandbreite wie eine Übertragung mit einem einzelnen Träger.

Die praktische Realisierung eines OFDM Systems, d.h. vor allem die Erzeugung bzw. die Auswertung der oft vielen hundert Subträger erfolgt im Normalfall rein digital. Die einzelnen Symbole einer OFDM Übertragung (d.h. die Summe aller modulierten Subträger) werden mit Hilfe einer entsprechenden IFFT aus den für dieses Symbol aktiven Konstellationspunkten der einzelnen Subträger berechnet. Das IFFT Ergebnis entspricht dem benötigten IQ-Basisbandsignal und wird anschließend mit Hilfe von zwei Digital-to-Analog (DA) Wandlern in die analoge Domäne übersetzt.

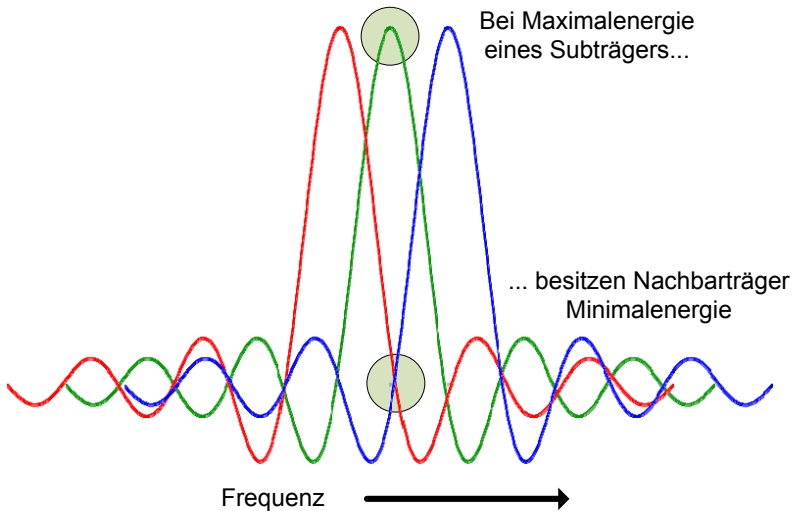


Abbildung 2.4.: OFDM Träger im Frequenzbereich

Dort wird es mit Hilfe eines IQ Modulators auf den Träger moduliert (vgl. auch Abschnitt 2.1.3). Für den Empfang wird der Träger zunächst mit einem IQ Demodulator ins Basisband herabgesetzt und mit Hilfe von zwei Analog-to-Digital (AD) Wandlern digitalisiert. Nachdem das empfangene OFDM Symbol eine FFT Berechnung passiert hat, lassen sich die Konstellationspunkte der einzelnen Subträger direkt aus dem FFT Ergebnis ablesen und dekodieren. Die Abbildungen 2.5 und 2.6 zeigen dieses stark vereinfachte Vorgehen einer OFDM Übertragung.

In praktischen Systemen müssen vor allem beim Empfänger noch einige weitere Signalverarbeitungs-komponenten eingesetzt werden. Dazu gehört allem voran eine Synchronisationseinheit. Diese hat die Aufgabe den Beginn einer Übertragung im Empfangssignal zu erkennen und die einzelnen OFDM Symbole für die weitere Verarbeitung durch eine FFT korrekt und ohne Symbolüberlappungen anzuschneiden. Eine weitere wichtige Aufgabe des Empfängers besteht in der Synchronisation der Oszillatoren. Die optimale Anordnung der Subträger wie in Abbildung 2.4 dargestellt, erfordert exakt getunte Oszillatoren. Bereits ein leichter

## 2. Grundlagen

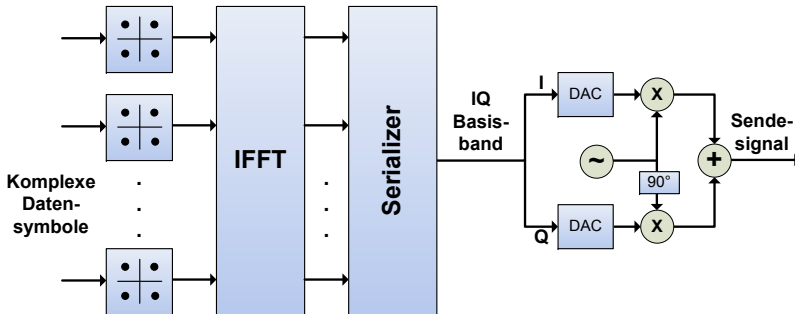


Abbildung 2.5.: Blockdiagramm eines stark vereinfachten OFDM Senders

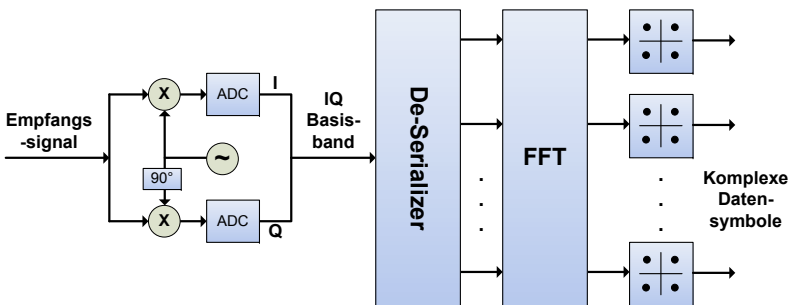


Abbildung 2.6.: Blockdiagramm eines stark vereinfachten OFDM Empfängers

Offset des Sampling- oder Lokoszillators führt zu einer Verschiebung des optimalen Subträgerabtastpunkts. Ein Subträger wird im Spektrum dann nicht mehr bei seinem Maximum abgetastet, sondern leicht daneben. Das hat zur Folge, dass die Trägerenergie nicht maximal ist und gleichzeitig Energie benachbarter Träger mit erfasst wird, welche sich als Stör-rauschen bemerkbar macht. Aus diesem Grund ist für den Betrieb eines OFDM Empfängers das Nachführen der Empfangsoszillatoren basierend auf Messergebnissen der Synchronisationseinheit und speziellen Pilot-Subträgern unabdingbar für einen hochqualitativen Empfang.

Im Folgenden werden einige wichtige Elemente eines OFDM Systems etwas genauer erläutert, da das in dieser Arbeit vorgestellte Kommunikationssystem auf OFDM basiert. Dazu gehört vor allem die praktische Realisierung einer FFT, sowie weitere unabdingbare Funktionseinheiten wie z.B. eine Autokorrelation.

### 2.2.2. Fast Fourier Transformation

Die Diskrete Fourier-Transformation (Diskreten Fourier-Transformation (DFT)) für den Vektor  $x$  der Dimension  $2n$  lautet:

$$f_m = \sum_{k=0}^{2n-1} x_k e^{-\frac{2\pi i}{2n} mk}$$

für  $m = 0, \dots, 2n-1$ . Diese Summe kann in zwei Teilsummen aufgeteilt werden, in denen jeweils die geraden bzw. die ungeraden  $x_k$  berechnet werden.

$$f_m = \sum_{k=0}^{n-1} x_{2k} e^{-\frac{2\pi i}{2n} m(2k)} + \sum_{k=0}^{n-1} x_{2k+1} e^{-\frac{2\pi i}{2n} m(2k+1)}$$

Ersetzt man die geraden Indizes durch

$$x_k^i = x_{2k}, k = 0, \dots, n-1$$

und die ungeraden Indizes durch

$$x_k^{ii} = x_{2k+1}, k = 0, \dots, n-1$$

lässt sich die letzte Gleichung auch folgendermaßen schreiben:

$$f_m = \sum_{k=0}^{n-1} x_k^i e^{-\frac{2\pi i}{n} mk} + e^{-\frac{\pi i}{n} m} \sum_{k=0}^{n-1} x_k^{ii} e^{-\frac{2\pi i}{n} mk}$$

Wird für  $2n$  eine Zweierpotenz gewählt, so kann diese Aufteilung in Teilsummen rekursiv immer weiter geführt werden bis man elementare FFT Operationen mit nur zwei oder vier Punkten erhält. Der Term  $e^{-\frac{\pi i}{n} m}$  wird auch als Drehfaktor bezeichnet, da er die Phase der zweiten Summe um einen von  $m$  abhängigen Wert dreht.

## 2. Grundlagen

---

Dieses sogenannte *Teile-und-Herrsche* Verfahren, d.h. die Unterteilung einer großen Transformation in viele Kleinere, bildet die Grundidee bei der Berechnung einer FFT. Eine 16-Punkt FFT kann beispielsweise durch 32 2-Punkt FFTs in vier Stufen oder effizienter durch 8 4-Punkt FFTs in zwei Stufen berechnet werden. Die elementaren 2-Punkt bzw. 4-Punkt FFT Algorithmen werden auch als *Radix-2* bzw. *Radix-4 Butterflies* bezeichnet und ihre Realisierung basiert auf der Multiplikation der komplexen Eingangswerte mit  $+1$ ,  $-1$ ,  $+j$  oder  $-j$  und einer anschließenden Summation wie in Abbildungen 2.7 und 2.8 dargestellt.

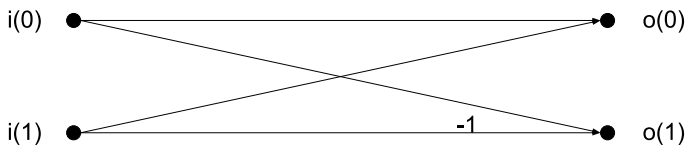


Abbildung 2.7.: FFT Radix-2 Butterfly

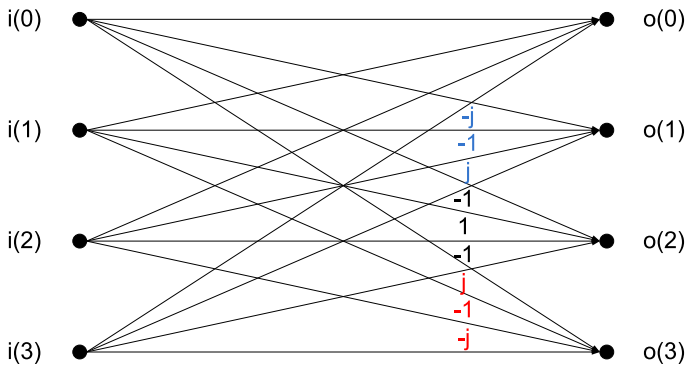


Abbildung 2.8.: FFT Radix-4 Butterfly

Die Kombination der einzelnen *Butterfly* Algorithmen zu größeren FFTs erfolgt dann wie in Abbildung 2.9 für das Beispiel einer 16 Punkte FFT dargestellt. Zunächst werden in einer ersten Stufe vier parallel und unabhängig voneinander arbeitende *Radix-4 Butterflies* auf den 16 Eingangswerten berechnet. Anschließend werden die ermittelten Zwischenergebnisse mit Drehfaktoren multipliziert und in einer zweiten Stufe erneut

in vier parallel arbeitenden *Radix-4 Butterflies* verknüpft. Zu guter Letzt werden die nun *bit-reverse* sortiert vorliegenden Berechnungsergebnisse der FFT in einem *Bit Reordering* bezeichneten Prozess umsortiert, so dass sie in der von der DFT bekannten Reihenfolge vorliegen.

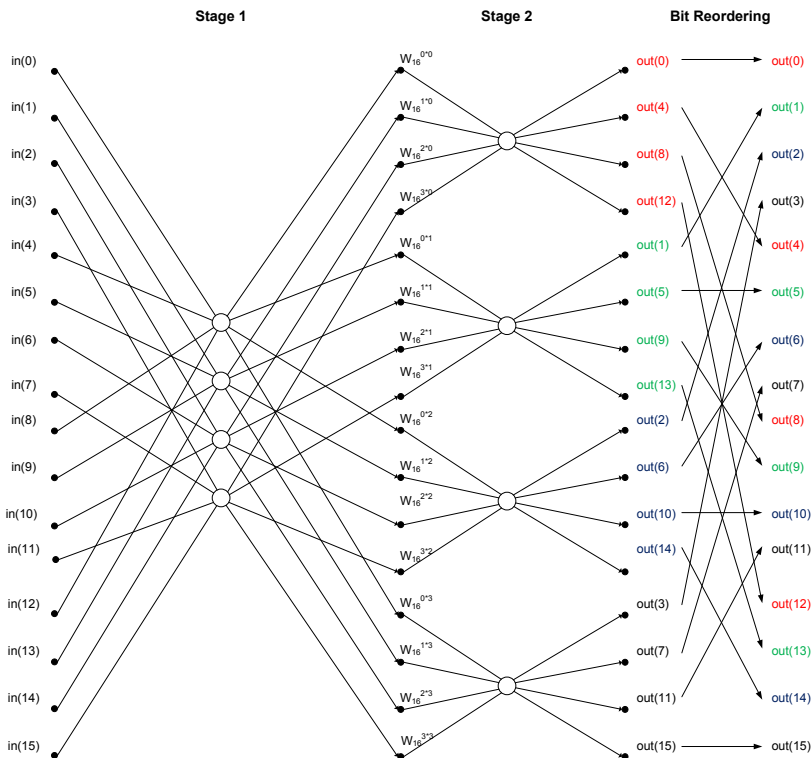


Abbildung 2.9.: 16 Punkt FFT in Radix 4 Struktur

Die FFT Struktur in Abbildung 2.9 wird auch als Decimation in Frequency (DIF) bezeichnet, da die Berechnung der Eingangswerte in immer kleinere, unabhängige Einheiten zerlegt wird, je weiter sie von der Zeitdomäne entfernt und zur Frequenzdomäne hin stattfindet. Die Berechnung der FFT kann jedoch genauso auch in einer Decimation in Time (DIT) Struktur stattfinden. Hier finden sich die kleineren Berechnungseinheiten in der

## 2. Grundlagen

Nähe der Zeitdomäne. Beide Strukturen sind grundsätzlich äquivalent bzgl. der Berechnung und der Komplexität. Wie wir jedoch noch sehen werden, können sich bei einer konkreten Realisierung Vorteile für die eine oder andere Struktur ergeben. Abbildung 2.10 zeigt den Vergleich zwischen einer DIT und einer DIF Realisierung.

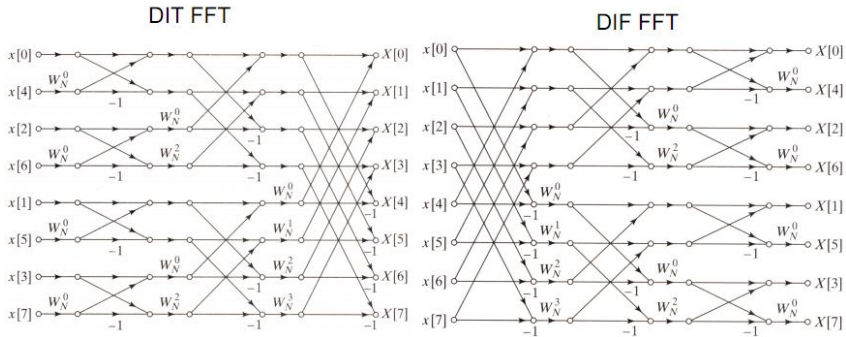


Abbildung 2.10.: Vergleich zwischen DIT und DIF FFT

Bisher wurde nur die Berechnung einer FFT betrachtet. Die Berechnung einer IFFT kann jedoch sehr einfach mit Hilfe einer FFT erfolgen, indem Real- und Imaginärteil sowohl am Eingang als auch am Ausgang der FFT vertauscht werden ([11]).

### 2.2.3. Autokorrelation basierte Synchronisation

Bei einer Korrelation in der Signalverarbeitung handelt es sich ganz allgemein um ein Maß zur Ähnlichkeit zwischen Signalen. Man unterscheidet zwischen der Kreuzkorrelation, welche zwei unterschiedliche Signale miteinander vergleicht, und der Autokorrelation, bei der ein Signal mit einer zeit-verschobenen Version von sich selbst verglichen wird.

Die Autokorrelation ist ein wichtiger Verarbeitungsschritt in den meisten OFDM Empfängern. Mit ihrer Hilfe wird im Empfangssignal eine Synchronisationssequenz detektiert, welche den Rahmenanfang einer Übertragung markiert. Die Synchronisationssequenz ist so aufgebaut,



dass sie aus mindestens zwei identischen Abschnitten besteht, so dass eine auf die entsprechende Verzögerung konfigurierte Autokorrelation die Sequenz eindeutig erkennen kann. Die Verwendung einer Kreuzkorrelation wäre hier weniger geeignet, da das Empfangssignal von der Übertragungstrecke in gewissem Umfang beeinflusst wird und so die Ähnlichkeit zwischen dem Empfangssignal und dem bekannten, gesendeten Referenzsignal nicht immer eindeutig gegeben ist.

Die Autokorrelationsfunktion (AKF) für zeitdiskrete Signale ist wie folgt definiert:

$$AKF(\tau) = \sum_{i=0}^{n-1} x_i x_{i-\tau}$$

Bei dieser Definition erfolgt die Summation über sämtliche aufgenommenen Samples. Im praktischen Einsatz innerhalb eines OFDM Empfängers ist dieses Vorgehen jedoch nicht praktikabel, denn es soll nicht das gesamte bereits empfangene Signal mit Hilfe der Autokorrelation untersucht werden, sondern lediglich ein kurzer Teil mit der Länge der Synchronisationssequenz. Aus diesem Grund erfolgt die Autokorrelation in diesem Fall lediglich über die letzten  $m$  Samples, wobei der Parameter  $m$  gleich der Länge der Synchronisationssequenz gesetzt wird:

$$AKF(\tau, m) = \sum_{i=n-m}^{n-1} x_i x_{i-\tau}$$

Ein größeres Korrelationsfenster wird nicht benötigt und wäre sogar kontraproduktiv, da in diesem Fall auch Bereiche des Empfangssignals mit in die Korrelationsberechnung einfließen würden, welche nicht zur Synchronisationssequenz gehören und mangels Ähnlichkeit mit dieser den Korrelationseffekt entsprechend reduzieren. Der Parameter  $\tau$  wird ebenfalls bereits zum Designzeitpunkt des Systems festgelegt. Er ergibt sich aus dem zeitlichen Abstand in Samples zwischen den beiden identischen Abschnitten der Synchronisationssequenz. Idealerweise wird er so gewählt, dass während der restlichen Übertragung des OFDM Signals keine Signalkonstellationen entstehen können, welche eine Signalwiederholung mit identischem zeitlichen Abstand aufweisen. In diesem Fall kann mit Hilfe der Autokorrelation die Synchronisationssequenz auch in stark verrauschten Signalen zuverlässig detektiert werden.

## 2. Grundlagen

---

Im Fall von OFDM Systemen kommt in der Regel die weiter oben beschriebene komplexe IQ-Basisbanddarstellung für die abgetasteten Samples zum Einsatz. In diesem Fall wird für den Multiplikator der konjugiert komplexe Wert verwendet. So ergibt sich für komplexe Werte die leicht abgewandelte AKF:

$$AKF(\tau, m) = \sum_{i=n-m}^{n-1} x_i^* x_{i-\tau}$$

Die Verwendung von komplexen IQ-Samples zur Berechnung der Autokorrelation bietet auch bei der Synchronisation Vorteile. Die Ähnlichkeit des verzögerten Signalteils mit dem aktuellen Empfangssignal ergibt sich, wie bei der reellen Variante oben, aus dem Betrag des Berechnungsergebnisses. Die Phase der AKF liefert nun jedoch zusätzlich ein Maß für die Ähnlichkeit der Phasenlage der einzelnen Samples beider verglichenen Signalteile. So ist es möglich, auch leichte Phasenabweichungen zwischen den Samples der beiden eigentlich identischen Abschnitte der Synchronisationssequenz zu ermitteln. Die ermittelte Abweichung stellt ein direktes Maß für den Offset des Lokaloszillators dar, denn, vorausgesetzt der Übertragungskanal verhält sich innerhalb der kurzen Zeitspanne der Synchronisationssequenz hinreichend stabil, bildet ein Frequenzoffset des Oszillators die einzige Quelle für eine langsam, aber kontinuierlich fortschreitende Phasenverschiebung.

Das hier beschriebene Vorgehen zur Detektion von Synchronisationssequenzen mit zwei oder mehr identischen Abschnitten mit Hilfe der Autokorrelation wurde ursprünglich in [35] beschrieben. Es liefert den exakten Zeitpunkt einer Synchronisationssequenz und zusätzlich ein Maß für die Abweichung des Lokaloszillators. Weiterhin kann bei bekanntem Sollabstand zwischen zwei Synchronisationssequenzen auch die Abweichung des Sampleoszillators sehr einfach ermittelt werden, indem der Empfänger die gemessene Anzahl an Symbolen zwischen den Sequenzen mit dem bekannten Sollwert vergleicht. Somit liefert das Verfahren dem Empfänger sämtliche zur Synchronisation benötigten Informationen woraus sich auch seine Popularität ergibt.

### 2.2.4. Pilottöne

Bei Pilottönen handelt es sich um ein weiteres Verfahren zur Ermittlung von Frequenzfehlern des Lokal- und Sampleoszillators. Im Gegensatz zu dem zuvor vorgestellten, auf Autokorrelation basierenden Verfahren nach [35] arbeitet es im Frequenz- anstatt im Zeitbereich. Die Grundidee besteht darin, einzelne Subträger des OFDM Spektrums mit bekannter Amplitude und Phase, einem sogenannten Pilotton, zu modulieren. Dabei können einzelne Subträger entweder kontinuierlich oder nur zeitweise einen Pilotton tragen oder die verwendeten Subträger variieren. Der Variantenvielfalt bei der konkreten Realisierung sind hier kaum Grenzen gesetzt und die gewählte Form hängt in der Regel sowohl vom erwarteten Übertragungskanal sowie von weiteren Systemanforderungen ab. Da dem Empfänger der Sollwert des Pilottons bekannt ist, kann er Phasen- und Amplitudenabweichungen der empfangenen Subträger kontinuierlich ermitteln und ausgleichen. Die Auswertung der Pilottöne kann jedoch erst nach der Berechnung der FFT im Empfänger erfolgen, daher eignet sich dieses Verfahren nicht für die initiale Synchronisation des Empfängers. Diese Aufgabe bleibt weiterhin Autokorrelation basierten Verfahren vorbehalten. Allerdings erlauben Pilotton basierte Verfahren eine kontinuierliche Überwachung, so dass ein einmal synchronisierter Empfänger im Idealfall keine erneute Synchronisationssequenz benötigt und die Datenbereiche der übertragenen Rahmen in der Praxis entsprechend lang ausfallen können.

## 2.3. Spezialisierte Funktionseinheiten

In den folgenden Unterabschnitten werden einige einfache, jedoch grundlegende Funktionseinheiten kurz erläutert, welche in nachfolgenden Kapiteln während des Designs des OFDM Systems an verschiedenen Stellen benötigt werden.

### 2.3.1. Numerically Controlled Oscillator

Ein numerisch gesteuerter Oszillator kann als digitales Pendant eines Voltage-Controlled-Oscillator (VCO), also eines spannungsgesteuerten Oszillators angesehen werden. Seine Aufgabe ist die Erzeugung einer Sinus-Schwingung (bzw. bei Bedarf auch einer um 90 Grad phasenverschobenen Kosinus-Schwingung) in Form eines digitalen Signalvektors. Die zu erzeugende Sollfrequenz wird ebenfalls durch einen digitalen Steuervektor vorgegeben und erstreckt sich in der Regel über einen sehr großen Bereich. Abbildung 2.11 zeigt den typischen Aufbau eines Numerically Controlled Oscillator (NCO).

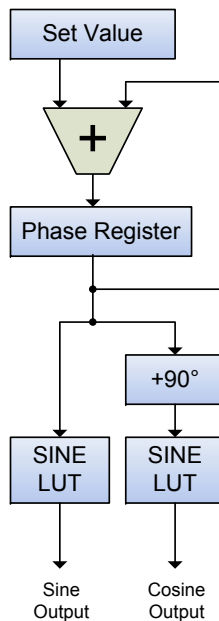


Abbildung 2.11.: Blockschaltbild eines NCO

Zentrales Element des NCO ist ein Phasenakkumulator, der zu jeder Zeit die aktuelle Phase der ausgegebenen Schwingung(en) enthält. Sein Wert wird mit jedem Takt um den Wert des Steuervektors inkrementiert oder

dekrementiert. Durch die Möglichkeit den Wert des Phasenakkumulators auch zu reduzieren, kann bei der Ausgabe eines Quadratursignals (also die kombinierte Ausgabe eines Sinus- und eines Kosinus-Signals) auch die Laufrichtung des Oszillators bestimmt werden. Das ist insbesondere bei der Signalverarbeitung im komplexen IQ-Basisband sehr nützlich, da so beispielsweise ein Frequenzoffset des Lokaloszillators sehr einfach korrigiert werden kann.

Der Wert des Phasenakkumulators wird schließlich mit Hilfe von Look-Up Tabellen, welche den Arcus-Sinus berechnen, in eine Sinus- bzw. Kosinus-Schwingung umgewandelt. Die maximal erzielbare Frequenz eines NCO ergibt sich aufgrund des Abtasttheorems zur Hälfte der Taktfrequenz. In der Praxis ist diese Frequenz jedoch nicht ganz erreichbar, da sich die Signalleistung der ausgegebenen Schwingungen bei Annäherung an diese Maximalfrequenz immer weiter reduziert. Die Genauigkeit der Look-Up Tabellen ist oftmals aufgrund von limitierender Speichergrößen auf Auflösungen im Bereich von 12 Bit begrenzt. Um trotzdem eine hohe Frequenzauflösung des NCO zu erhalten, wird der Phasenakkumulator häufig mit 32 oder mehr Bits betrieben, während lediglich dessen höchstwertigste Bits an die Lookup-Tabellen weitergereicht werden.

### 2.3.2. Direct Digital Synthesis

Die direkten digitalen (Schwingungs-) Synthese (Direct Digital Synthesis (DDS)) bildet eine Erweiterung eines NCO. Die digitale Ausgabe der Sinus- bzw. Kosinus-Schwingungen des NCO wird mit Hilfe von DA Wandlern von der digitalen in die analoge Domäne übersetzt. Bei einer DDS handelt es sich also um einen sehr feingranular, digital steuerbaren Oszillator. Durch diese Eigenschaft ist er hervorragend geeignet, um sehr geringe Frequenzoffsets in der analogen Domäne auszugleichen. In der hier vorgestellten Arbeit basiert beispielsweise die Korrektur der Samplingoszillatorfrequenz der Teilnehmeranschlüsse auf der Verwendung einer DDS. Abbildung 2.12 zeigt ein um die Komponenten einer DDS erweitertes Blockschaltbild eines NCO. Der wesentliche Unterschied besteht in dem Vorhandensein zusätzlicher DA Wandler.

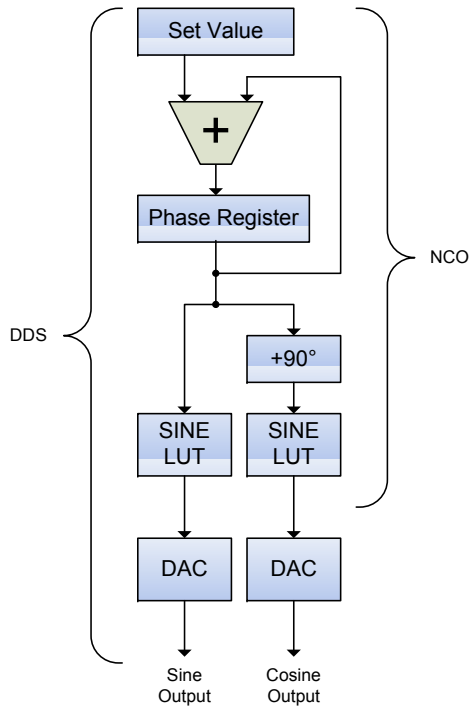


Abbildung 2.12.: Um DDS Komponenten erweiterter NCO

### 2.3.3. Phase Locked Loop

Ein sog. Phasenregelkreis bezeichnet ein häufig eingesetztes System zur phasenstarken Frequenzvervielfachung. Es besteht im Wesentlichen aus den vier, in Abbildung 2.13 eingezeichneten Komponenten.

Ein *Referenz-Oszillator* erzeugt eine sehr genaue Referenzfrequenz, welche um einen gewissen Faktor vervielfältigt werden soll. Gleichzeitig arbeitet ein VCO zunächst frei schwingend und erzeugt eine Schwingung, die in etwa der später gewünschten Ausgangsfrequenz entspricht. Diese Schwingung wird mit Hilfe eines Frequenzteilers auf eine in der Regel deutlich tiefere Frequenz herunter geteilt, welche in etwa der Refe-

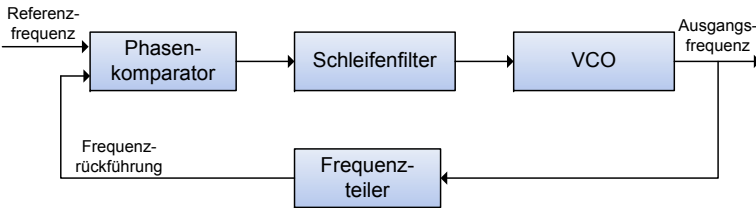


Abbildung 2.13.: Blockschaltbild einer PLL

renzfrequenz entsprechen sollte. Ein Phasenkomparator vergleicht nun die Referenzfrequenz mit der herunter geteilten Ausgangsfrequenz und liefert ein Ausgangssignal, dessen Höhe proportional zum Phasenversatz der beiden Schwingungen ist. Der anschließende sog. *Schleifenfilter* integriert das Ausgangssignal des Phasenkomparators und steuert schließlich den VCO. So ergibt sich ein Regelkreis und die Ausgangsfrequenz des VCO wird so nachgeführt, dass sie exakt und phasenstarr einem Vielfachen der Referenzfrequenz entspricht.

PLLs werden immer dann eingesetzt, wenn hochfrequente und gleichzeitig sehr frequenzstabile Schwingungen erzeugt werden müssen, die nicht direkt durch die Verwendung entsprechend genauer Oszillatoren bereitgestellt werden können. Die häufig genutzten sehr frequenzstabilen Quarzoszillatoren lassen sich beispielsweise nur bis zu etwa 100 MHz einsetzen. Weiterhin erlauben PLLs die Erzeugung weiterer, phasenstarr gekoppelter und von Jitter bereinigter Frequenzen aus rückgewonnenen Schwingungen beispielsweise in Empfängern und kommen somit auch häufig in Empfangsschaltkreisen zum Einsatz.

In dieser Arbeit wird eine PLL zur Vervielfältigung der Ausgangsfrequenz der Sampletakt erzeugenden DDS verwendet, da diese selbst nicht in der Lage ist, die hohe benötigte Frequenz im GHz Bereich zu erzeugen.

### 2.3.4. CORDIC Algorithmus

Ein Coordinate Rotation Digital Computer (CORDIC) wird zur Berechnung trigonometrischer Funktionen sowie Logarithmen, Exponential-

## 2. Grundlagen

---

funktionen und der Quadratwurzel eingesetzt. In dieser Arbeit liegt sein Anwendungsgebiet vor allem in der Bestimmung von Phase und Amplitude komplexer Basisbandsamples und bei der Berechnung der Sinus- und Kosinusfunktion. Der Algorithmus wurde 1959 zum ersten Mal veröffentlicht und wird somit bereits entsprechend lange in Computersystemen eingesetzt, wie z.B. in den 80x87-Floating Point Unit (FPU) Koprozessoren von Intel. Der iterative Algorithmus basiert auf in Hardware einfach zu realisierenden Additions-, Subtraktions- und Schiebeoperationen. Grundidee ist es, einen Vektor im zweidimensionalen Koordinatensystem zu drehen. Wird als Startvektor der Einheitsvektor gewählt und so lange gedreht, bis ein vorgegebener Winkel erreicht ist, erhält man den Sinus bzw. Kosinus des Winkels als Endvektor. Dieser Betriebsmodus wird auch als Rotationsmodus bezeichnet. Alternativ kann im Vektormodus statt dem Einheitsvektor auch ein beliebiger Vektor vorgegeben werden, welcher so gedreht wird, dass die  $y$ -Komponente des Vektors minimal wird. Am Ende enthält die  $x$ -Komponente des Vektors den Betrag und der zurückgelegte Winkel entspricht der Phase.

Die effiziente Rotation des Vektors basiert auf dem Matrix-Vektor-Produkt:

$$\begin{pmatrix} x_e \\ y_e \end{pmatrix} = \begin{pmatrix} \cos \Theta & -\sin \Theta \\ \sin \Theta & \cos \Theta \end{pmatrix} * \begin{pmatrix} x_s \\ y_s \end{pmatrix}$$

Alternativ kann sie auch wie folgt mit Hilfe des Tangens beschrieben werden:

$$\begin{pmatrix} x_e \\ y_e \end{pmatrix} = \cos \Theta * \begin{pmatrix} 1 & -\tan \Theta \\ \tan \Theta & 1 \end{pmatrix} * \begin{pmatrix} x_s \\ y_s \end{pmatrix}$$

Die Drehung um  $\Theta$  wird als Linearkombination von Teildrehungen um die Teilwinkel  $\alpha_i$  ausgeführt:

$$\Theta = \sum_i \sigma_i * \alpha_i$$



$\sigma_i$  kann die Werte +1 und -1 annehmen und gibt somit das Vorzeichen des addierten Winkels vor. Die einzelnen Teilwinkel werden geschickt zu  $\tan \alpha_i = 2^{-i}$  gewählt und die Werte für  $\tan \alpha_i$  in einer Tabelle vorberechnet. Die Länge der Tabelle gibt die Anzahl  $n$  der Drehiterationen des Algorithmus vor und bestimmt so die Genauigkeit der Berechnung.

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \prod_{i=0}^n \cos \alpha_i \begin{pmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{pmatrix} * \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

Da der Wert von  $\cos \alpha_i$  unabhängig vom Vorzeichen des Winkels ist, kann er zu der Konstanten  $K = \prod_{i=0}^n \cos \alpha_i \approx 0,60725$  berechnet werden. Dieser Wert wird auch als der CORDIC-Gain bezeichnet.

Aus der obigen Formel folgt der iterative Ansatz:

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i 2^{-i} y_i \\ y_{i+1} &= y_i + \sigma_i 2^{-i} x_i \\ z_{i+1} &= z_i - \sigma_i \arctan 2^{-i} \end{aligned}$$

Die Multiplikation von  $\sigma_i 2^{-i} y_i$  bzw.  $\sigma_i 2^{-i} x_i$  lässt sich durch einfache Schiebeoperationen durchführen und die Operation  $\sigma_i \arctan 2^{-i}$  ist in der Tabelle hinterlegt.

Der Startvektor wird in jedem Iterationsschritt der Berechnung um den Winkel  $\alpha_i$  rotiert. Der Winkel startet im ersten Schritt bei 45 Grad und wird in jedem weiteren Schritt aufgrund der Wahl von  $\tan \alpha_i = 2^{-i}$  halbiert. Das Vorzeichen des Rotationswinkels in jedem Schritt ergibt sich aus der Zielvorgabe des gewählten Modus. Im Rotationsmodus wird es so gewählt, dass sich die aktuelle Winkelsumme dem Zielwinkel nähert. Im Vektormodus wird das Vorzeichen so gewählt, dass sich die  $y$ -Komponente dem Wert 0 nähert. Nach der letzten Iteration kann das Berechnungsergebnis je nach gewähltem Modus aus  $x_n$ ,  $y_n$  und  $z_n$  entnommen werden.

### 2.3.5. PID-Regler

Bei einem Proportional-Integral-Derivative Regler handelt es sich um eine Rückkopplungsstruktur für Regelkreise, die sehr häufig für einfache Regelaufgaben in den unterschiedlichsten Bereichen eingesetzt wird. Ein Regler sorgt selbständig dafür, dass ein Parameter eines technischen Prozesses auch beim Auftreten von Störgrößen möglichst genau einem Sollwert entspricht. Häufig kann der Parameter zwar beobachtet, jedoch nicht direkt beeinflusst werden. In diesem Fall wird der Parameter durch die Manipulation einer anderen, geeigneten Stellgröße des Prozesses indirekt so beeinflusst, dass die Regelabweichung minimiert wird. Die Struktur eines Proportional-Integral-Derivative (PID) Reglers ist in Abbildung 2.14 dargestellt.

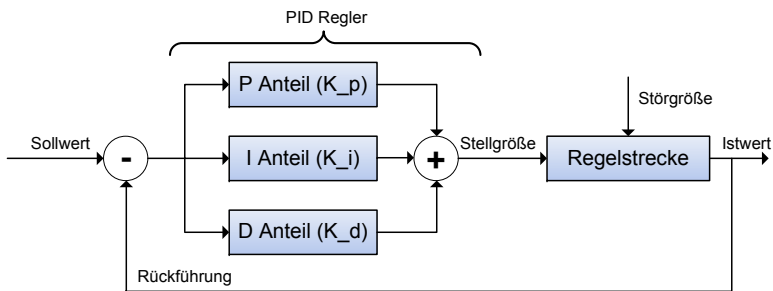


Abbildung 2.14.: Prinzipieller Aufbau eines PID-Reglers

Der PID Regler berechnet zunächst die Regelabweichung aus der Differenz zwischen dem Sollwert und dem Istwert des zu regelnden Parameters. Anschließend existieren drei Pfade, von denen der Regler seinen Namen erhalten hat:

- Der *proportionale Pfad (P)* führt die Regelabweichung direkt, proportional um den Faktor  $K_p$  verstärkt, auf die Stellgröße zurück. Damit wirkt er unmittelbar, hinterlässt jedoch in der Regel eine kleine Abweichung vom Sollwert, da ohne Regelabweichung auch keine Stellgröße ungleich Null möglich ist.
- Der *integrale Pfad (I)* führt eine zeitliche Integration der Regelabweichung, um den Faktor  $K_i$  verstärkt, auf die Stellgröße zurück.

Jegliche Regelabweichung führt nach einer gewissen Zeit zu einer ausreichenden Stellgröße, so dass auf Dauer bei statischen Verhältnissen keine Regelabweichung verbleibt.

- Der *differentielle Pfad* ( $D$ ) führt die Änderungsrate der Regelabweichung, um den Faktor  $K_d$  verstärkt, auf die Stellgröße zurück. Er ist nur zusammen mit einem P und/oder einem I Pfad sinnvoll einsetzbar. Er wird verwendet, um vorausschauend auf die Stellgröße zu wirken wenn eine starke Änderung der Regelabweichung auftritt und kann so die Reaktionszeit der Regelung reduzieren. Der differentielle Pfad sollte sparsam eingesetzt werden, da er die Regelung schnell unruhig oder sogar instabil macht.

Diese drei Pfade wirken in einem PID Regler parallel. Indem der Verstärkungsfaktor ( $K_p$ ,  $K_i$  oder  $K_d$ ) eines Pfades und damit auch dessen Einfluss auf die Stellgröße zu Null gesetzt wird, können mit derselben Reglerstruktur auch andere häufig eingesetzte Varianten wie  $P$ ,  $PI$ ,  $PD$  oder  $I$  Regler realisiert werden.

Die in dieser Arbeit auftretenden Regelkreise beschränken sich auf die Nachführung der Oszillatoren an den Teilnehmeranschlüssen und sind damit vergleichsweise einfach. Sie entsprechen praktisch der Phasenregelung innerhalb einer PLL und sind in der Regel als reine  $I$ -Regler ausgeführt. Trotzdem wurden in der Realisierung des Kommunikationssystems dieser Arbeit vollwertige PID Reglerstrukturen eingesetzt, um für alle Fälle gewappnet zu sein.

## 2.4. Field Programmable Gate Array

### 2.4.1. Aufbau

Der typische grundsätzliche Aufbau eines FPGA gestaltet sich relativ einfach. Aufgabe eines FPGA ist es, eine beliebige digitale Schaltung durch die Programmierung mit einer sog. Konfiguration abzubilden. Dazu muss es sowohl über eine Möglichkeit verfügen, beliebige logische Funktionen abzubilden, als auch Speicherelemente in Form von D-FlipFlops zur Verfügung zu stellen. Weiterhin müssen diese beiden grundlegen-

## 2. Grundlagen

---

den Elemente flexibel untereinander verschaltbar sein und es muss eine ausreichende Anzahl der Elemente zur Realisierung der gewünschten digitalen Schaltung zur Verfügung stehen.

Die Realisierung beliebiger Logikfunktionen basiert in der Regel auf der Verwendung von Lookup-Tabellen. Eine heute übliche Tabellengröße ist 64 Bit. D.h. eine Lookup-Tabelle besitzt sechs Adresseingänge und einen digitalen Ausgang. Auf diese Weise lassen sich sämtliche binären Logikfunktionen beschreiben, welche bis zu sechs digitale Eingangssignale besitzen. Moderne FPGAs besitzen häufig viele weitere Eigenschaften, um die Ressourcenausnutzung und Taktgeschwindigkeit der Schaltungen zu steigern. Beispielsweise lassen sich bei dem in dieser Arbeit verwendeten FPGA Typ *Virtex-6* Lookup-Tabellen unterteilen, so dass aus einer 64 Bit Tabelle zwei 32 Bit Tabellen entstehen. So existieren pro Tabelle zwar nur noch fünf digitale Eingänge, dafür erhält man jedoch zwei anstelle von einem binären Ausgang. Weitere Optimierungen erfolgen beispielsweise mit Hilfe spezieller sog. Carry-Chains. Dabei handelt es sich um fest verdrahtete Logik zur Realisierung von Addierern, Subtrahieren oder Zählern. Diese Funktionen werden erfahrungsgemäß besonders häufig bei der Realisierung digitaler Schaltungen benötigt. Tabelle 2.1 fasst die wichtigsten Ressourcen der beiden in dieser Arbeit einsetzbaren FPGAs zusammen.

Baustein	LUTs	BRAMs (36 Kb)	Multiplizierer	GTX	GTH	IO Pins
XC6VHX380T	239040	768	864	48	24	720
XC6VHX565T	354240	912	864	48	24	720

Tabelle 2.1.: Ressourcenübersicht der verwendete Virtex-6 FPGAs

Eine weitere Eigenschaft moderner FPGAs besteht in der Einbettung auch größerer, festverdrahteter Funktionsgruppen. Dazu gehören vor allem Speicherblöcke, welche meist wenige Kilobytes pro Block fassen und häufig zusätzlich als asynchroner First-In First-Out Memory (FIFO) dienen können sowie Multiplizierer bzw. komplette Multiply-Accumulate Einheiten. Aber auch ganze Prozessorblöcke und Speichercontroller sind nicht selten anzutreffen.

Auch im Bereich der externen Interfaces eines FPGA hat sich in den letzten Jahren einiges getan. Standen früher lediglich einfache Input/Output (IO) Blöcke zur Verfügung, die lediglich eine Pegelwandlung zwischen der IO Spannung eines Pins und der internen Logik durchführten, so enthalten moderne IO Blöcke weitere Logik. Dazu gehören etwa Double Data Rate (DDR) FlipFlops zur Ansteuerung externer Baustein mit schnellen IO Schnittstellen, einstellbare Signalverzögerungen und Serializer/Deserializer (SERDES) Blöcke, welche die Aus- bzw. Eingabe von Signalen mit bis zu einem GBit/s durch jeden normalen IO Pin ermöglichen.

Ein weiterer, moderner, in dieser Arbeit besonders häufig verwendeter IO Block ist der sog. Multi-Gigabit-Transceiver (MGT). Er erlaubt das Senden und Empfangen von Signalen bei vielen GBit/s. Da er hier so häufig eingesetzt wird, folgt im nächsten Abschnitt eine detaillierte Beschreibung.

### 2.4.2. Multi-Gigabit Transceiver

Multi-Gigabit-Transceiver (MGTs) finden in dieser Arbeit an vielen verschiedenen Stellen Verwendung. Sei es zur Anbindung von AD und DA Wandlern, zur direkten Kommunikation zwischen verschiedenen FPGAs oder für die Realisierung von 10G Ethernet Schnittstellen. Aus diesem Grund soll im Folgenden eine kurze Einführung in die grundsätzliche Funktionsweise dieser Transceiver gegeben werden.

Ein MGT ist eine Kommunikationsschnittstelle, die in der Lage ist, serielle Signale bei einer Bitrate von mehr als 1 GBit/s zu senden und zu empfangen. Die ersten MGTs wurden von Xilinx unter dem Namen *RocketIO* in der *Virtex-II Pro* FPGA Serie eingeführt. Sie waren in der Lage, Datenraten bis zu 3.125 GBit/s zu verarbeiten. Zum Start der *Virtex-5* Serie wurde ein einheitliches Namensschema eingeführt: GTP Transceiver (Virtex-5, Spartan-6) arbeiten bei bis zu 3.125 GBit/s, GTX Transceiver (Virtex-5, Virtex-6) erlauben Datenraten von bis zu 6.25 GBit/s, GTH Transceiver (Virtex-6) schaffen 11 GBit/s und GTZ Transceiver (Virtex-7) ermöglichen bis zu 26 GBit/s. Für FPGAs von Altera existieren ähnliche Unterteilungen der verfügbaren Transceiver in unterschiedliche Geschwindigkeitsklassen.

## 2. Grundlagen

---

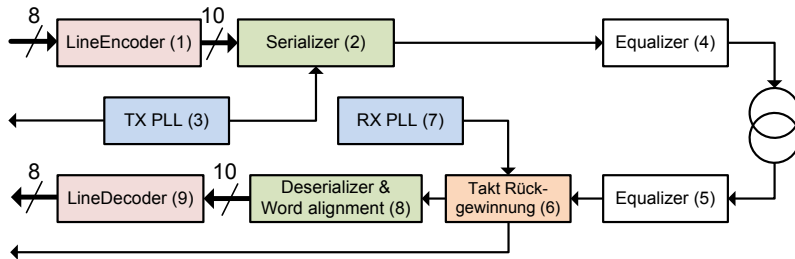


Abbildung 2.15.: Vereinfachtes Blockdiagramm eines MGT

Abbildung 2.15 illustriert die Schritte, welche typischerweise in einer einfachen Datenübertragung mittels eines MGT involviert sind.

- Zu Beginn der Übertragung wird ein paralleler Datenstrom, der bei einer Wortfrequenz von deutlich unter 1 GHz vorliegt mit Hilfe eines geeigneten Linecodes, z.B. 8B10B, in das Sendesignal konvertiert (1). Der Linecode entfernt normalerweise sämtliche DC Komponenten aus dem Sendesignal, um eine Übermittlung des Signals über größere Distanzen zu ermöglichen, da hier typischerweise aufgrund von Kapazitäten oder Übertragern im Signalpfad nur AC Signale erlaubt sind. Weiterhin garantiert der Linecode eine ausreichend hohe Anzahl an Pegelwechseln im Sendesignal, wodurch eine zuverlässige Taktrückgewinnung beim Empfänger des Signals ermöglicht wird. Die meisten verfügbaren MGTs erlauben auch den Lineencoder zu umgehen, so dass eine geeignete Kodierung im FPGA selbst durchgeführt werden kann. So lassen sich auch sehr spezielle oder neue Codes nutzen.
- Das Sendesignal wird nun mit Hilfe eines Sendetakts serialisiert (2). Der Sendetakt wird von einer PLL (3) aus einem Referenztakt generiert. Die PLL kann typischerweise sehr flexibel konfiguriert werden und deckt so einen großen Frequenzbereich ab.
- Nach einer optionalen Vorverzerrung (4) im analogen Frontend des Transceivers, wird das Signal vom FPGA ausgegeben.

- Beim Empfänger passiert des Empfangssignal zunächst einen weiteren Equalizer (5), der es von Frequenzverzerrungen durch den Übertragungskanal befreit.
- Im nächsten Schritt werden die Signalfanken ausgewertet und eine Taktrückgewinnung wird auf Basis eines Empfangstakts durchgeführt (6). Der Empfangstakt wird, wie auch der Sendetakt, von einer PLL (7) aus einem Referenztakt gewonnen. Dieser Schritt ist notwendig, da in den meisten Fällen die Frequenz des Empfangstakts nicht exakt der des Sendetakts entspricht, der an einem entfernten FPGA basierend auf einem eventuell leicht abweichenden Referenztakt generiert wird.
- Anschließend wird das empfangene Datensignal zu einem parallelen Datenstrom bei einer Wortrate deutlich unter 1 GHz deserialisiert (8). Außerdem findet in diesem Schritt die Zuordnung der einzelnen Empfangsbits zu den Wortgrenzen statt. Dazu sind typischerweise spezielle Bitsequenzen vom Linecode definiert, die zur Erkennung der Wortgrenzen genutzt werden.
- Im letzten Schritt wird schließlich noch der Linecode von den Empfangsdaten entfernt und der originale Datenstrom wird wieder sichtbar. An dieser Stelle sollte angemerkt werden, dass die bei einem Kommunikationspartner empfangene Datenrate in der Regel nicht exakt der gesendeten Datenrate des selben Kommunikationspartners entspricht. Dies resultiert aus leichten Frequenzunterschieden der genutzten Referenztakts der beiden Kommunikationspartner.

MGTs werden in der Regel verwendet, um Datenlinks im Bereich von mehreren GBit/s zu betreiben. Bekannte Beispiele dazu sind PCI-Express, 10G Ethernet oder SATA. MGTs in FPGAs können typischerweise sehr flexibel konfiguriert werden, so dass sie eine große Vielfalt unterschiedlichster Protokolle unterstützen. Eine Liste der Schnittstellen, welche beispielsweise von einem GTX Transceiver abgedeckt werden, findet sich im GTX Handbuch auf [44] auf Seite 117. Diese sehr große Flexibilität erlaubt auch die Realisierung von maßgeschneiderten Protokollen. Typischerweise können sämtliche bei der Verarbeitung von Sende- und Empfangsdaten involvierten Blöcke eines MGT sehr detailliert konfiguriert werden. Die wichtigsten dieser Blöcke bilden dabei die analoge Frontendelektronik, die Takterzeugung, die Taktrückgewinnung am Empfänger

## 2. Grundlagen

---

sowie diverse digitale Schaltungsteile für die Serialisierung, Deserialisierung, Linekodierung und Blockausrichtung. Aufgrund der hohen Anzahl an Verarbeitungsschritten während der Übertragung, besitzt ein Datensignal, welches ein MGT Interface passiert, eine deutlich größere Latenz verglichen mit einem Signal, welches mit Hilfe eines normalen IO Pins übermittelt wird. Diese Tatsache muss bei der Umsetzung von Protokollen unter Verwendung von MGTs beachtet werden.

Ein weiterer wichtiger Punkt, der bei der Nutzung von MGTs beachtet werden muss, ist die Datenintegrität. Auch wenn bekannt sein sollte, dass jeder Datenlink potentiell anfällig für Übertragungsfehler ist, haben sich langsame und einfache Links, die beispielsweise auf normalen FPGA IO Pins basieren, als sehr stabil und zuverlässig erwiesen. Für diese Links kann im Normalfall davon ausgegangen werden, dass keine Bitfehler auftreten und somit keine Fehlererkennung oder gar Korrektur benötigt wird, solange das Signal lediglich über kurze Strecken, z.B. innerhalb der Grenzen einer Platine, übermittelt wird. Bei der Verwendung von Multi-Gigabit-Links (MGLs) ändert sich dieses Verhalten jedoch. Beispielsweise wird bei 10 GBit/s das impedanzangepasste Routing der Leiterbahnen auf einer Platine so wichtig, dass bereits eine unbeabsichtigte Berührung des Bords zu Übertragungsfehlern führt. Das bedeutet, wenn MGLs zuverlässig eingesetzt werden sollen, müssen immer auch Fehlererkennungs- und Korrekturmechanismen umgesetzt werden.

Für den zu verwendenden Linecode existiert eine große Auswahl. Beginnend von einem 4B5B Code, welcher beispielsweise bei Fast Ethernet Verwendung findet, stößt man schnell auf den 8B10B Code, welcher z.B. in PCI-Express, SATA oder HDMI eine weite Verbreitung findet. Hochgeschwindigkeitsverbindungen wie z.B. 10G, 40G oder 100G Ethernet verwenden typischerweise einen 64B66B Code, welcher ein 64 Bit breites Datenword auf ein 66 Bit breites Codewort umsetzt und somit einen geringeren Overhead verglichen mit den zuvor genannten Codes aufweist.



## 3. Stand der Technik

Die Grundidee der OFDM Übertragung, auf der diese Arbeit fußt, ist bereits seit Ende der 1960er bekannt und wurde in einem Patent veröffentlicht [6]. Einige Jahre später, 1971, folgte schließlich ein Artikel [42], welcher das Verfahren in der Art und Weise beschreibt, wie es heute in der Regel verwendet wird. Aufgrund fehlender Rechenleistung sollte es jedoch noch eine ganze Weile dauern, bis OFDM in praktischen Systemen zum Einsatz kommt. Die erste breite Anwendung wurde Mitte der 1990er Jahre mit der Einführung von drahtgebundenen DSL Systemen verwirklicht. Von da an verbreitete sich OFDM wegen seiner guten Signalausbreitungseigenschaften vor allem auch im Bereich der drahtlosen Anwendungen. Heute ist es nicht mehr wegzudenken, es basieren Standards für die digitale Ausstrahlung von Radio und Fernsehen (DAB, DVB-T) genauso auf einer OFDM Übertragung wie moderne Wireless LAN Varianten (802.11n, 802.11ac, 802.11ad) und Mobilfunksysteme (LTE). Der Artikel *The History of Orthogonal Frequency-Division Multiplexing* von Stephen B. Weinstein [41] fasst die Geschichte des Modulationsverfahrens zusammen.

OFDM bzw. dessen Erweiterungen stellen heute das beste bekannte Verfahren bzgl. ihrer spektralen Effizienz in praktischen Anwendungsszenarien dar. Es erreicht nahezu die theoretische Obergrenze der Bitrate eines Übertragungskanals nach dem Shannon-Hartley-Gesetz [38]. Daher ist es nicht verwunderlich, dass das Verfahren auch heute noch von vielen Forschungsgruppen aktiv verbessert und weiterentwickelt wird. In jüngster Zeit fand es daher, neben draht- und funkgebundenen Kommunikationssystemen, auch Einzug in Forschungsaktivitäten zur Verbesserung der Bandbreitenausnutzung in optischen Kommunikationssystemen. Gerade hier ist die Umsetzung besonders schwierig, da die Datenraten in der Optik in der Regel deutlich höher liegen als in den beiden anderen Domänen. Aus diesem Grund finden sich auch hier noch keinerlei kommerziell

eingesetzte Systeme die auf OFDM basieren und die Forschungsaktivität ist entsprechend hoch.

Da auch bei dieser Arbeit maximale Performanz des realisierten OFDM Systems im Vordergrund steht, ergeben sich folglich auch die größten inhaltlichen Überschneidungen mit anderen Aktivitäten aus dem Umfeld optischer Netzwerke.

## 3.1. OFDM in optischen Kommunikationssystemen

Aufgrund der sehr hohen Bandbreite von Glasfasern wurden lange Zeit nur sehr einfache Modulationsformate wie beispielsweise On-Off-Keying (OOK) für optische Kommunikationssysteme betrachtet. Das lag zum einen an einer ausreichend hohen Bandbreite der optischen Netze, so dass eine effizientere Modulationstechnik schlicht nicht notwendig war. Zum anderen erfordern die extrem hohen Bandbreiten bereits bei einfachen Modulationsformaten hohe Rechenleistungen bei den angeschlossenen Kommunikationspartnern, so dass komplexere Formate, zu denen insbesondere auch OFDM zählt, die benötigte Rechenleistung für die Signalverarbeitung auf extrem hohe Werte steigen lässt, welche erst seit kurzem mit modernster Hardware erreichbar ist.

Durch die in jüngster Zeit stetig ansteigenden Bandbreitenanforderungen und eine Entwicklung der Netzwerktopologie hin zu Fiber-To-The-Home (FTTH) oder Fiber-To-The-Building (FTTB) spielt auch eine effiziente spektrale Nutzung des optischen Kanals zunehmend eine Rolle. Die Nutzung von OFDM bzw. OFDMA, deren Vorteile bereits durch drahtgebundene und drahtlose Anwendungen bekannt sind, wurde somit auch für den Bereich optischer Kommunikationssysteme interessant [4]. Tatsächlich stand in vielen Forschungsaktivitäten jedoch die eigentliche Übermittlung eines OFDM Signals über optische Kanäle im Vordergrund, da es sich beispielsweise aufgrund eines im Vergleich zu Einträgerverfahren deutlich höheren Peak-to-Average-Power-Ratio (PAPR) stark von etablierten optischen Modulationstechniken unterscheidet. Auch an die Linearität der Übertragungsstrecke stellt OFDM höhere Anforderungen

als einfachere Verfahren, so dass sich viele Veröffentlichungen aus dem Bereich optisch übermittelter OFDM Signale auf die Realisierung geeigneter Übertragungstrecken sowie den dazugehörigen analog optisch- bzw. elektrischen Frontends beschäftigen [28], [36]. Das eigentliche Sendesignal wird häufig vorberechnet und mit Hilfe eines Arbitrary Waveform Generator (AWG) im Experiment eingespeist, was sicherlich auch im hohen benötigten Rechenaufwand und der damit verbundenen komplizierten Umsetzung sowie der teuren benötigten Hardware begründet liegt.

Trotzdem existieren durchaus einige Veröffentlichungen, in denen zumindest das OFDM Signal für die optische Übertragung in Echtzeit generiert wird [37]. Jedoch wird auch hier das empfangene Signal typischerweise mit Hilfe eines Oszilloskops aufgezeichnet und anschließend offline ausgewertet. Veröffentlichungen zu kompletten Übertragungssystemen mit für optische Netzwerke typischen Bandbreiten im zweistelligen GBit/s Bereich, welche sowohl ein OFDM Signal in Echtzeit generieren als auch in Echtzeit empfangen, finden sich dagegen selten [18], [21], [17]. Das dürfte vor allem an dem im Vergleich zu einem OFDM Sender deutlich höheren Implementierungsaufwand sowie der benötigten Rechenleistung liegen. Entsprechend findet man bei der Veröffentlichung von Empfängern für Hochgeschwindigkeitssysteme häufig mehr oder weniger starke Vereinfachungen. Beispielsweise können die Taktsignale für den Sample- und Lokaloszillator für Sender und Empfänger aus dem selben Oszillator gewonnen werden, so dass eine aufwändige Zeit- und Frequenzsynchronisation am Empfänger eingespart werden kann. Die Taktsignale des Senders werden dem Empfänger dann entweder mittels dedizierter Leitungen oder zusätzlicher optischer Träger auf der Glasfaser zugeführt. Ein solches System taugt zwar, um die Übermittlung eines OFDM Signals genau zu studieren, jedoch ist es nicht für eine Anwendung außerhalb der Labors geeignet.

Eine weitere Einschränkung typischer Forschungssysteme für optische OFDM basierte Kommunikation ist häufig die Limitierung der Übertragung auf der Ebene der Trägerkonstellationen. Das bedeutet, es werden komplexe Symbole für die einzelnen Subträger vorgegeben und überprüft, ob ein korrekter Empfang statt fand bzw. der Fehlervektor wird im Empfänger ausgewertet [27]. Höhere Kommunikationsschichten wie beispielsweise ein Interleaving der Sendedaten oder eine Forward Error

Correction (FEC) zur Sicherung der Übertragung werden typischerweise nicht betrachtet. Sie sind aber gerade in OFDM Systemen integraler Bestandteil, da das Mehrträgerverfahren nur so die im Vergleich zu einfacheren Modulationstechniken höhere Zuverlässigkeit und Datenrate erzielen kann. Soweit überschaubar, ist die hier vorgestellte Arbeit die erste, welche ein komplettes OFDM basiertes Kommunikationssystem mit einer Bandbreite vergleichbar zu aktuellen optischen Time Division Multiple Access (TDMA) basierten Systemen behandelt, und dabei reale Nutzdaten in Form von Ethernet Paketen übermittelt.

## 3.2. Rechenarchitekturen für OFDM basierte Kommunikationssysteme

Wie bei den allermeisten digitalen System existieren auch für ein OFDM basiertes Kommunikationssystem diverse Realisierungsmöglichkeiten.

Eine typische Realisierung basiert, wie auch bei vielen anderen signalverarbeitenden Systemen, auf dem Einsatz von klassischen digitalen Signalprozessoren. Besonders zu Beginn des Einsatzes von Mehrträgerverfahren kam diese Variante aufgrund fehlender Hardwareimplementierungen und noch recht moderater Datenraten häufig zum Einsatz. Schnell wurden jedoch wegen steigender Datenraten und aufwändigerer Übertragungsverfahren die Anforderungen an die signalverarbeitende Hardware immer höher, so dass die OFDM Prozessierung heute in der Regel mit Hilfe spezialisierter Application-Specific Integrated Circuits (ASICs) sehr energieeffizient und kostengünstig realisiert wird.

Parallel zu dieser Entwicklung werden aufgrund der zunehmenden Integration von *DSP* Funktionen in normale Prozessoren sowie dem Vorstoß von Graphics Processing Units (GPUs) in die Signalverarbeitungsdomäne auch viele Mischformen für die Prozessierung ermöglicht. Eine Architektur, welche in besonderem Maße auf eine gemischte Hardware/Software Verarbeitung von OFDM Signalen optimiert ist, wird in [24] vorgestellt. Sie erlaubt einerseits den Einsatz performanter Hardwarebeschleuniger in Form von intelligenten Koprozessoren, während trotzdem die Flexibilität einer Softwarelösung weitgehend beibehalten wird.

An die Performanz reiner und optimierter Hardwarelösungen reichen jedoch auch solche gemischten Implementierungen nicht heran. Da in dieser Arbeit die Performanz des entwickelten Systems an erster Stelle steht, wurde hier auf eine reine hardwarebasierte Lösung gesetzt.

## 3.3. Funktionseinheiten

### 3.3.1. FFT Berechnung

Unabhängig von der Betrachtung des Gesamtsystems zur OFDMA basierten Hochgeschwindigkeitsdatenübertragung lohnt es sich, auch die Entwicklung einiger elementarer Komponenten des Systems zu betrachten. Aufgrund ihres hohen Ressourcenverbrauchs zählt dazu allen voran die FFT.

Sie kommt in vielen Kommunikationssystemen zum Einsatz, spielt darüber hinaus jedoch auch eine große Rolle in vielen weiteren Bereichen. Die Palette reicht dabei von verschiedenen Anwendungen in der Messtechnik über Kompressionsverfahren wie beispielsweise dem bekannten MP3-Format bis hin zu Aufgaben in der Computeralgebra. Mit der Weiterentwicklung der Rechnertechnik ergeben sich immer neue Einsatzgebiete für immer schneller arbeitende FFT Algorithmen. Es lohnt sich daher auch ein Blick in Veröffentlichungen anderer Bereiche.

Bei der Fast Fourier Transform handelt es sich um einen effizienten Algorithmus zur Berechnung einer Diskreten Fourier-Transformation (DFT). Der FFT Algorithmus wurde 1965 von James Cooley und John W. Tukey vorgeschlagen [8]. Er nutzt Redundanzen in der Berechnung der DFT, um die Anzahl der benötigten Multiplikationen und Additionen zu minimieren.

Aufgrund der immensen Bedeutung des Algorithmus in vielen unterschiedlichen Domänen existieren viele Veröffentlichungen, die sich mit einer optimalen Realisierung einer FFT in unterschiedlichen Hardwarestrukturen befassen. Glenn D. Bergland präsentierte bereits 1969 eine Übersicht zu verschiedenen Implementierungsansätzen des Algorithmus

[5]. Er vergleicht im wesentlichen vier prinzipielle Architekturen in seiner Untersuchung:

- Ein sequentieller Prozessor, in der eine arithmetische Einheit sämtliche Berechnungen übernimmt
- Eine Berechnungseinheit für jede FFT Stufe
- Eine Berechnungseinheit für jede Berechnung innerhalb einer FFT Stufe
- Eine Berechnungseinheit für jede Berechnung innerhalb der gesamten FFT (*Array Analyzer*)

Die meisten Veröffentlichungen der letzten Jahre konzentrieren sich auf den ersten Ansatz basierend auf einer sequentiellen Prozessorarchitektur. Der Grund dafür liegt vermutlich in der hohen Flexibilität dieses Ansatzes, im geringen Ressourcenverbrauch und in der Tatsache, dass für viele Anwendungen eine moderate Performanz der FFT Berechnung genügt.

Es existieren diverse Möglichkeiten zur Optimierung der Architektur des sequenziellen Prozessors. A. M. Despain beispielsweise schlägt einen Austausch der Drehfaktormultiplikation durch einen CORDIC Algorithmus vor [9], [10]. Durch die Einführung von Pipelining kann dieser Ansatz weiter verbessert werden und zu schnelleren Implementierungen führen wie in [2] gezeigt wird. Weitere Veröffentlichungen wie beispielsweise in [32] schlagen die Verwendung von unterschiedlichen Strukturen zur Berechnung der *Butterflies* vor. Auch existieren neben der Verwendung eines CORDIC Algorithmus noch weitere Wege zur Optimierung der Drehfaktormultiplikationen [34]. Eine gute Übersicht über unterschiedliche Architekturen für die Berechnung einer FFT mittels sequentieller Prozessoren findet sich in [46].

Aufgrund verschiedener Entwicklungen in vielen Bereichen, in denen FFTs eingesetzt werden stieg vor allem in den letzten Jahren das Interesse an Implementierungen, die einen sehr hohen Durchsatz erlauben. Die übliche Methode, um eine solche Steigerung der Berechnungsleistung zu erzielen ist die Verwendung mehrerer sequentieller Prozessoren zusammen mit einem Kontrollmechanismus, welcher die einzelnen Einheiten koordiniert und die Berechnungsdaten entsprechend aufteilt [47], [33].

Ein interessanter Ansatz zur „Berechnung“ einer FFT wird in [19] vorgestellt. Hier erfolgt der Übergang zwischen Zeit- und Frequenzbereich nicht durch die Abarbeitung eines Algorithmus, sondern mit Hilfe eines geeignet strukturierten optischen Netzwerks. Ihre Architektur entspricht im Prinzip der eines *Array Analyzer*, allerdings arbeitet sie kontinuierlich und analog und ist aufgrund des hohen Hardwareaufwandes nur für geringe Punktezahlen geeignet. Wegen ihrer kontinuierlichen und analogen Natur erzielt diese rein optisch arbeitende FFT einen extrem hohen Durchsatz. Ein Nachteil ist jedoch, dass ihre Berechnungsgenauigkeit deutlich geringer ausfallen dürfte als die von digitalen Implementierungen und auch eine Integration in die weitere (digitale) Signalverarbeitung ist äußerst schwierig.

Zur Erzielung höchster Datenraten wurde in dieser Arbeit eine voll parallele Implementierung des FFT Algorithmus (*Array Analyzer*) als Startpunkt gewählt. Diese Implementierung wurde weiter optimiert, um den erzielten Durchsatz zu maximieren und parallel dazu den Ressourcenverbrauch zu minimieren. Daher zählt sie mit zu den schnellsten FFT Implementierungen, welche jemals veröffentlicht wurden.

#### 3.3.2. Zeit- und Frequenzsynchronisation

Eine besondere Herausforderung bei der Entwicklung von OFDM Systemen stellt die Realisierung einer sehr genauen Zeit- und Frequenzsynchronisation dar. Da sich die einzelnen Subträger im Spektrum überlappen, müssen die Frequenzen der Sample- sowie Lokaloszillatoren von Sender und Empfänger sehr exakt übereinstimmen, damit die einzelnen Subträger wie in Abbildung 2.4 dargestellt tatsächlich bei ihrem Leistungsmaximum im Spektrum vom Empfänger abgetastet werden. Ein Frequenzoffset des Lokaloszillators führt zu einer Verschiebung der Abtastpunkte im Spektrum, während ein Offset des Sampleoszillators zu einer Stauchung bzw. Dehnung führt. Die Synchronisation erfolgt heute typischerweise in mindestens zwei Schritten. Zunächst wird das Empfangssignal im Zeitbereich betrachtet und entsprechende Korrekturwerte ermittelt, so dass eine korrekte Überführung in den Frequenzbereich mit Hilfe einer FFT Berechnung überhaupt ermöglicht wird. Im zweiten Schritt kann das empfangene Signal im Frequenzbereich analysiert wer-

den, wodurch genauere bzw. kontinuierlich ermittelte Korrekturwerte ermittelt werden können [20].

Das Standardverfahren, welches heute zur Synchronisation im Zeitbereich verwendet wird, wurde 1997 von Timothy M. Schmidl and Donald C. Cox veröffentlicht [35]. Es basiert im Wesentlichen auf einer Autokorrelation zur Detektion einer speziellen Synchronisationssequenz innerhalb des Empfangssignals. Im Frequenzbereich kommen in der Regel diverse Pilottöne zum Einsatz, welche je nach Anforderungen des Übertragungssystems nahezu beliebig auf unterschiedliche Subträger und Symbole verteilt werden können [7].

Aufgrund der Bedeutung einer genauen Frequenzsynchronisation für die korrekte Funktion von OFDM Systemen und der vielfältigen Möglichkeiten diese zu erreichen, existieren verhältnismäßig viele Veröffentlichungen zu diesem Thema. Dazu gehören beispielsweise auch Varianten, die auf den ersten Blick etwas kurios erscheinen wie zum Beispiel eine in [12] beschriebene Auswertung von Pilottönen im Zeitbereich. Auch durch das in jüngster Zeit verstärkte Aufkommen von MIMO Systemen ergeben sich neue Herausforderungen für die Zeit- und Frequenzsynchronisierung [48].

#### 3.3.3. Fehlerkorrektur

Eine weiterer integraler Bestandteil eines jeden praktisch einsetzbaren OFDM Systems bilden Mechanismen zur Fehlerkorrektur. Hier werden meist verschiedene Verfahren kombiniert, um eine maximale Absicherung der übermittelten Daten zu erreichen. Allerdings gilt wie bei den FFT Algorithmen auch hier, dass der Einsatzbereich von Fehlerkorrekturmechanismen sehr weit gestreut ist. Sie spielen bei den allermeisten Datenübertragungssystemen eine entscheidende Rolle und kommen auch bei vielen modernen Datenspeichern zum Einsatz, seien es Festplatten, CDs, DVDs oder Flashspeicher. Entsprechend existieren auch zu den Fehlerkorrekturverfahren viele Veröffentlichungen aus den unterschiedlichsten Anwendungsbereichen.

Grundsätzlich existieren zwei Varianten zur Absicherung von Daten gegen Bitfehler. Zum einen besteht die Möglichkeit, die Integrität der Daten lediglich zu überprüfen und sie im Fall eines festgestellten Fehlers neu an-



zufordern. Diese Verfahren benötigen in der Regel verhältnismäßig wenig Rechenleistung, da beim Empfänger lediglich eine Prüfsumme gebildet werden muss. Sie erfordern jedoch eine vollständige Neuübermittlung eines unter Umständen großen Datenblocks falls nur ein einzelner Bitfehler auftritt, wodurch entsprechende Latenzen entstehen.

Zum anderen besteht die Möglichkeit, jedem Datenblock generell eine gewisse Menge Redundanz hinzuzufügen, so dass eine gewisse Anzahl von Bitfehlern vom Empfänger auch ohne Neuansforderung des Datenblocks korrigierbar ist. Typischerweise wird dieses zweite, als Forward Error Correction (FEC) bezeichnete Verfahren, für die unteren Kommunikationsschichten eingesetzt, da gerade in der Funk- und Weitbereichskommunikation immer wieder einzelne Bitfehler auftreten und so eine ständige Neuübermittlung des Datenblocks vermieden wird. Das erste Verfahren kommt dagegen in der Regel in den höheren Schichten des OSI-Modells zum Einsatz, falls die FEC einer unteren Schicht versagt hat. Eine FEC benötigt jedoch deutlich mehr Rechenleistung, da hier die Daten aus der Redundanzinformation wiederhergestellt werden müssen.

Da das in dieser Arbeit vorgestellte Kommunikationssystem in den untersten Schichten des OSI-Modells angesiedelt ist, erfordert seine praktische Realisierung entsprechend den Einsatz einer FEC. Es wird jedoch nicht im Detail auf die Realisierung von FEC Algorithmen für Hochgeschwindigkeitsdatenübertragungen eingegangen, da sich mit diesem Thema problemlos einige weitere Promotionen füllen lassen. Stattdessen kommt eine Realisierung von der Internetplattform *Opencores* zum Einsatz [31]. Es soll hier jedoch aufgezeigt werden, welche Schwierigkeiten auch die Realisierung eines performanten FEC Decoders mit sich bringt. In [29] wird die Entwicklung einer Soft-Decision FEC basierend auf Low-Density Parity-Check (LDPC) Codes diskutiert.



## 4. Systemarchitektur

Bevor mit der Realisierung eines Kommunikationssystems begonnen werden kann, muss es zunächst genau spezifiziert werden. Insbesondere wenn wesentliche Teile des Systems nicht in Software sondern in Form eines Hardwarebeschleunigers ausgeführt sind, ist eine exakte Planung von Beginn an sehr wichtig, denn die Implementierung und eventuelle spätere Änderungen an den Beschleunigern sind sehr aufwändig und kostenintensiv.

Basierend auf den Anforderungen an das Kommunikationsnetz werden die Systemparameter definiert. Aufbauend darauf kann anschließend die Architektur des Downlinks und des Uplinks in Form von Blockdiagrammen erstellt und optimiert werden. Hier wird die genaue Aufgabe und Funktionsweise jedes einzelnen Blocks exakt festgelegt, so dass während der späteren Implementierungsphase des Systems im Idealfall keine Unklarheiten auftreten können. Sind die Signalverarbeitungsblöcke definiert, kann die Spezifikation für die Einbettung von Nutzdaten in die übertragenen OFDM Symbole erfolgen. Hier wird definiert, wie genau die zu übermittelnden Ethernet Pakete in den Symbolstrom eingebettet werden.

Selbstverständlich wurde die Spezifikation während der Bearbeitung immer wieder den Bedürfnissen der Implementierung oder anderen Faktoren in einem iterativen Prozess angepasst. Zum bessern Verständnis erfolgen die Ausführungen in diesem Kapitel jedoch sequenziell, so dass im weiteren Verlauf dieses Kapitels die Spezifikationen in den zuvor beschriebenen Bereichen dargestellt werden.

### 4.1. Anforderungen

#### 4.1.1. Einsatzbereich des Kommunikationssystems

Ziel dieser Arbeit ist die exemplarische Realisierung eines OFDM basierten Hochgeschwindigkeitskommunikationssystems mit Hilfe von moderner rekonfigurierbarer Hardware. Die dabei anvisierte Bandbreite des Gesamtnetzes von etwa 25 GHz übersteigt bei weitem die Möglichkeiten heutiger funk- oder kupferbasierter Kommunikationsnetze, so dass die einzige realistische Anwendung im Bereich der optischen Datenkommunikation liegt.

Betrachtet man die verschiedenen Typen optischer Netze, wird schnell klar, dass sich ein komplexes und bandbreiteneffizientes Mehrträgerverfahren wie OFDM vor allem im Bereich der Zugangsnetze gewinnbringend einsetzen lässt. Während im Fall von Weitverkehrsnetzen eine Erhöhung der Datenrate unter Zuhilfenahme zusätzlicher Fasern oder Wellenlängen meist problemlos möglich ist, ist dies bei der Anbindung von z.B. Gebäuden oder Mobilfunkstationen häufig nicht so einfach. Meist ist hier nur eine einzige Glasfaser verlegt, welche viele Teilnehmer in Form einer Punkt-zu-Mehrpunkt Kommunikationsverbindung gemeinsam mit einer Vermittlungsstelle verbindet. Weiterhin ist aufgrund der hohen Anzahl von Teilnehmeranschlüssen eine Lösung wünschenswert, die nur mit einer Wellenlänge arbeitet um Kosten für unterschiedliche Laser und optische Filter einzusparen [39]. Da die Glasfasern der Zugangsnetze außerdem bereits viele weitere Dienste wie z.B. Kabelfernsehen übertragen, ist auch die verfügbare Bandbreite nicht beliebig groß, so dass sich der Einsatz eines effizienten Modulationsverfahrens hier besonders anbietet ([22]).

Die Verwendung von OFDM bietet durch die Aufteilung des Gesamtspektrums in Subträger jedoch noch eine weitere Möglichkeit, Kosten bei den Teilnehmeranschlüssen einzusparen: Bei geeigneter Auslegung des Systems reicht es aus, wenn jeder Teilnehmeranschluss nur einen Bruchteil der Bandbreite des Gesamtnetzwerks verarbeitet, so dass diese entsprechend kostengünstig konstruiert werden können. So muss an jeder Optical Network Unit (ONU) nur ein Bruchteil der Gesamtbandbreite prozessiert werden und entsprechend kostengünstig können diese kon-

struiert sein. Diese Einsatzvariante von OFDM wird auch als OFDMA bezeichnet. Bestehende optische Zugangsnetze wie z.B. Gigabit Passive Optical Network (GPON) setzen stattdessen typischerweise auf TDMA. Die Teilnehmeranschlüsse dieser Systeme müssen somit die gesamte Netzwerkbandbreite prozessieren wie sie von der Vermittlungsstelle bereitgestellt wird. Mit Hilfe von OFDMA sind sogar Systeme denkbar, bei denen die Gesamtbandbreite des Netzwerks an der Vermittlungsstelle bei Bedarf nahtlos ausgebaut wird, ohne dass die Teilnehmeranschlüsse davon beeinflusst werden.

Die sich aus dem Einsatzbereich des Kommunikationssystems ergebenden Anforderungen an das zu entwickelnde OFDM System können wie folgt zusammengefasst werden:

- *Hoher Durchsatz des Gesamtnetzwerks ( $\geq 50$  GBit/s):*  
Das neu entwickelte System soll mit bereits bestehenden Systemen konkurrieren können bzw. diese idealerweise übertreffen.
- *Einsatz einer ausreichend hohen Anzahl von Subträgern ( $\geq 256$ ):*  
Damit bei hoher Gesamtbandbreite des Netzwerks die zu verarbeitende Bandbreite an den Teilnehmeranschlüssen moderat bleibt, müssen entsprechend viele Subträger verfügbar sein, um die benötigte Granularität zu gewährleisten. Da pro Teilnehmer neben einigen Subträgern zur Datenkommunikation auch Träger für Pilottöne etc. benötigt werden, sind für ein sinnvolles System deutlich mehr als 64 Subträger notwendig. Eine effiziente Berechnung der für ein OFDMA System benötigten FFT an der Vermittlungsstelle auf Basis von Radix-4 Butterflies ist neben einer Punktezahl von 64 erst wieder mit 256 Punkten möglich. Aus diesem Grund wird hier eine minimale Subträgeranzahl von 256 als sinnvoll erachtet.
- *Entwicklung eines geeigneten Medium Access Control (MAC) Protokolls:*  
Jeder Teilnehmer des Netzwerks kann nur auf die Subträger in seinem aktuell gewählten Bandbreitenfenster direkt und simultan zugreifen. Für den Zugriff auf Subträger außerhalb dieses Fensters müssen Oszillatoren am Teilnehmeranschluss umgestimmt werden, was eine gewisse Zeitspanne benötigt und somit einen unterbrechungsfreien Wechsel auf Subträger außerhalb dieses Fensters unmöglich macht. Das eingesetzte MAC Protokoll, welches die Zu-

weisung der Teilnehmer auf verschiedene Subträgerbereiche regelt, muss diesen Umstand entsprechend berücksichtigen.

- *Minimierung von Out-Of-Band (OOB) Störaussendungen:*  
Da das Kommunikationsnetzwerk für Glasfaserstrecken in Zugangsnetzwerken konzipiert wird welche bereits weitere Dienste transportieren, muss eine Störung dieser Dienste unbedingt ausgeschlossen werden.

Diese vier Punkte stellen die wesentlichen Anforderungen an das zu entwickelnde OFDMA System dar und fließen direkt in die Spezifikation der Systemparameter ein.

### 4.1.2. Analog-Elektronisches und Optisches System

Weitere Anforderungen an die Systemspezifikation ergeben sich aus dem Aufbau der optischen und analog-elektronischen Netzwerkkomponenten. Diese begrenzen die verfügbare Signalbandbreite und führen zu einer Dämpfung des Signals auf langen Strecken. Auch die vom digitalen Systemteil zu kompensierenden Phasen- und Frequenzoffsets der Lokal- und Sampleoszillatoren entstehen hier.

In Abbildung 4.1 ist die topologische Struktur eines typischen passiven optischen Zugangsnetzwerks dargestellt, welches für ein Punkt-zu-Mehrpunkt (P2MP) Kommunikationssystem genutzt werden kann. Startend von einer zentralen Vermittlungsstelle (Optical Line Termination (OLT)) führt eine sog. *Feeder Fiber* zu einem ersten optischen Splitter. Von diesem Punkt aus führen mehrere Glasfasern zu weiteren Splittern oder bereits direkt zu Teilnehmeranschlüssen (ONU). Die Splitter sind nicht wellenlängenselektiv ausgelegt sondern für die gesamte auf der Faser nutzbare Signalbandbreite durchlässig. So hat jeder Teilnehmeranschluss grundsätzlich die Möglichkeit, auf sämtlichen Frequenzen zu operieren. Abhängig von der Position einer ONU im Netzwerk und ihrer Entfernung zum OLT ergeben sich jedoch unterschiedliche Signaldämpfungen und somit verschiedene SNRs. Entsprechend müssen in einem OFDM System die Modulationsformate der Subträger für jeden Teilnehmer einzeln anpassbar sein.

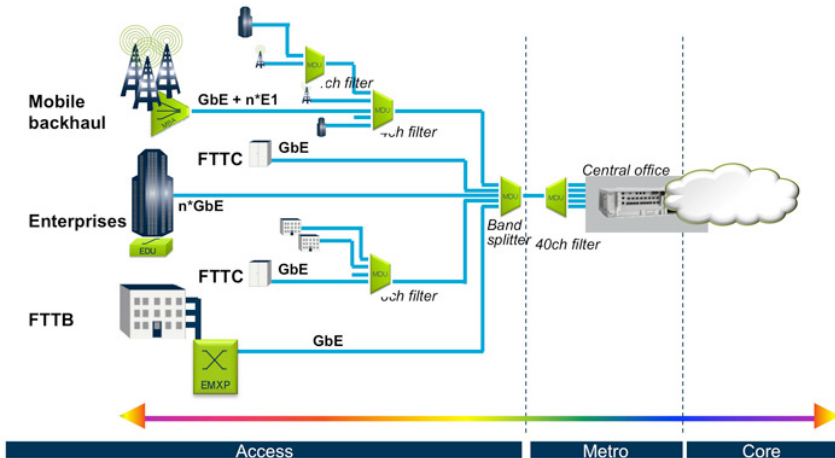


Abbildung 4.1.: Topologische Struktur eines optischen Zugangsnetzwerks [40]

Eine Kommunikationsverbindung auf OFDM Basis hat deutlich höhere Anforderungen an die verwendete Übertragungsstrecke als beispielsweise eine einfache, auf OOK basierende Modulation. Die Strecke muss sich über weite Bereiche linear verhalten und das Trägersignal in Phase und Amplitude variabel sein. Zum Test des in dieser Arbeit entwickelten OFDM Systems wurde alle wesentlichen Teile der in Abbildung 4.2 dargestellten Übertragungsstrecke aufgebaut.

Das in der Vermittlungsstelle digital erzeugte Basisbandsignal des Downlinks wird mit Hilfe zweier DA Wandler in ein analoges Signal umgesetzt und anschließend direkt mit Hilfe eines optischen Mach Zehnder IQ Modulators auf den Träger des Downlink Lasers moduliert. Das modulierte Trägersignal (*DS-Sig*) wird zusammen mit einem um 25 GHz versetzten Hilfsträger (*DS-LO*) auf die Feeder-Fiber (*ODN*) geleitet und so zu den Teilnehmeranschlüssen transportiert. Hier mischt der modulierte Träger mit dem um 25 GHz versetzten Hilfsträger, so dass am Ausgang der Photodiode das ursprünglich gesendete Basisbandsignal bei einem Frequenzoffset von 25 GHz abgegriffen werden kann. Dieses Signal wird nun mit Hilfe zweier kaskadierter elektrischer Mischer ins Basisband

#### 4. Systemarchitektur

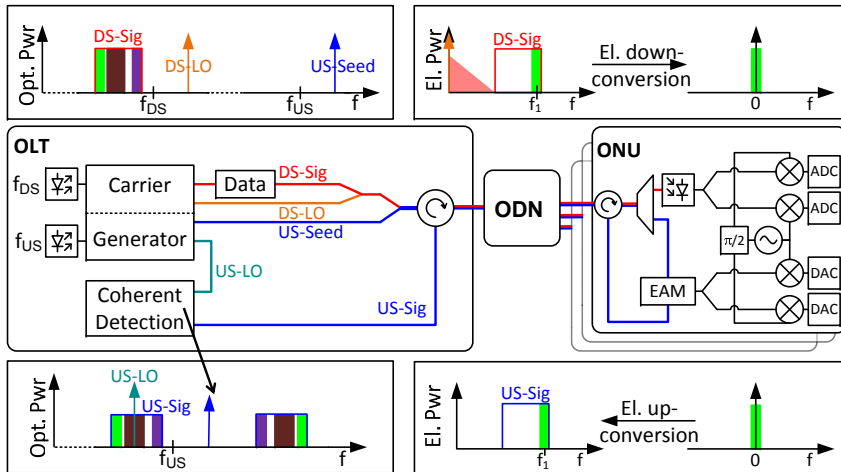


Abbildung 4.2.: Optisches Testübertragungsnetz

herabgesetzt und auf 1.5625 GHz bandbegrenzt, so dass es von zwei AD Wandlern digitalisiert werden kann und schließlich bereit ist für die weitere digitale Verarbeitung. Dieses Empfangsverfahren wird auch als *Remote Heterodyne Detection* bezeichnet.

Zur Realisierung des Uplinks wird an der Vermittlungsstelle ein weiterer optischer Hilfsträger (*US-Seed*) erzeugt und zu den Teilnehmeranschlüssen gesendet. Das hier mittels zweier DA Wandler erzeugte Basisbandsignal wird ähnlich dem Downlink Empfänger auf einen 25 GHz Träger hochgemischt und genutzt um den Uplink-Hilfsträger der Vermittlungsstelle mittels eines Electro-Absorption Modulator (EAM) zu modulieren und wieder zurückzusenden. Da alle Teilnehmer dasselbe Trägersignal der Vermittlungsstelle modulieren, addieren sich, vorausgesetzt die elektrischen Oszillatoren in den Uplink Mischerstufen an den ONUs sind korrekt gelockt, die von verschiedenen Teilnehmern erzeugten Spektren orthogonal und es bildet sich wieder ein OFDM Signal (*US-Sig*) an der Vermittlungsstelle.

Zusammenfassend lässt sich festhalten, dass die Aufgabe der analogen Systemteile darin besteht, im Sender das von den digitalen Signalverar-



beitungseinheiten erzeugte IQ-Basisbandsignal direkt auf das von einem Laser erzeugte Trägersignal zu modulieren, dieses modulierte Signal im Feld zu übertragen und schließlich am Empfänger wieder in ein IQ-Basisbandsignal zurückzuverwandeln. Wichtig ist, dass für die Lokal- und Samplingoszillatoren in der Vermittlungsstelle sowie bei den Teilnehmern eine gemeinsame Taktquelle sowohl für den Uplink als auch für den Downlink genutzt wird. So kann sichergestellt werden, dass nach der Synchronisation der Oszillatoren für eine Übertragungsrichtung unmittelbar auch die entgegengesetzte Richtung synchronisiert ist. Der Aufwand zur Synchronisation des Gesamtnetzes wird damit erheblich reduziert, wie später noch gezeigt wird.

### 4.1.3. Eingesetzte Hardware

#### 4.1.3.1. Komponentenauswahl

Bei der Planung eines Kommunikationssystems, welches die Grenzen des derzeit technisch Machbaren ausloten soll, kommt den verwendeten Hardwarekomponenten ein ganz besonderes Augenmerk zu. Die höchsten Anforderungen werden dabei sicherlich an die DA und AD Wandler sowie die Signalprozessierung der Vermittlungsstelle gestellt, welche die gesamte Netzwerkbandbreite von etwa 25 GHz abdecken müssen. Aus diesem Grund wird bei der Selektion der Hardware zunächst die Vermittlungsstelle ins Zentrum der Betrachtungen gestellt.

Aufgrund der IQ-Basisbandsignalisierung mit einer Bandbreite von 25 GHz werden jeweils zwei DA bzw. AD Wandler mit einer Bandbreite von 12.5 GHz und damit mindestens 25 GSa/s benötigt. Während der Planungsphase dieser Arbeit kam dazu nur der Hersteller Micram mit seinen VEGA Digital-to-Analog Convertern (DACs) und Analog-to-Digital-Convertern (ADCs) in Frage [25], [26]. Diese arbeiten bei einer Auflösung von 6 Bit mit bis zu 30 GSa/s und bieten somit sogar noch ein wenig Bandbreitenreserve. Deutlich kritischer ist jedoch die Auflösung von nur 6 Bit: Ein hohes PAPR ist typisch für OFDM (vgl. Kapitel 2), so dass der Dynamikbereich der OFDM Signale während der digitalen Verarbeitung mittels Clipping entsprechend begrenzt werden muss. Andererseits reduziert eine geringere Auflösung der Wandler die benötigte digita-

#### 4. Systemarchitektur

---

le Bandbreite zur Anbindung an die Signalverarbeitungseinheiten auf ein handhabbares Maß. An der Vermittlungsstelle kommen also jeweils zwei Micram VEGA DAC30 und zwei VEGA ADC30 zur Wandlung der IQ-Signale zum Einsatz.

Die zweite wichtige Hardwarekomponente ist zweifelsohne das an der Vermittlungsstelle verwendete FPGA. Es hat die Aufgabe die von den DA und AD Wandlern benötigten bzw. gelieferten digitalen Datenströme zu verarbeiten. Unabhängig von den im FPGA vorhandenen Logikressourcen steht zunächst einmal die am FPGA verfügbare IO-Bandbreite im Vordergrund: Zur Anbindung eines einzelnen Wandlers der Vermittlungsstelle an das FPGA werden 24 serielle digitale Links, sog. MGLs mit einer Bandbreite von jeweils 6.25 GBit/s benötigt. Dabei werden immer vier Links in den Wandlern mit Hilfe von Multiplexern zu einem Bit zusammengefasst. Da jeder Wandler insgesamt über eine Auflösung von 6 Bit verfügt, werden in Summe 24 Datenlinks benötigt. Allerdings werden zur Verarbeitung des IQ-Signals jeweils zwei DA bzw. AD Wandler benötigt deren Signale in der digitalen Domäne gemeinsam verarbeitet werden müssen. Daher benötigt das verwendete FPGA mindestens 48 dieser MGLs. In der Praxis sind die Links zur Erhöhung der Störsicherheit in Form von differenziellen Low Voltage Differential Signaling (LVDS) Signalen ausgeführt, so dass für die Anbindung von zwei Wandlern 96 Leitungen und damit auch Pins am FPGA benötigt werden.

Zum Zeitpunkt der Planung des Kommunikationssystems existierten lediglich zwei FPGA Typen aus der Virtex-6 Serie des Herstellers Xilinx, welche die geforderte IO Bandbreite erfüllten: Sowohl der Typ XC6VHX380T als auch XC6VHX565T stellen in der 1924 Pin-Version insgesamt 48 sog. GTX Transceiver zur Verfügung. Dabei handelt es sich um MGTs, welche serielle Datenströme mit einer Geschwindigkeit von bis zu 6.5 GBit/s simultan senden und empfangen können. Somit ist der parallele Betrieb beider DACs und beider ADCs der Vermittlungsstelle an FPGAs dieses Typs möglich.

Weiterhin stellen die beiden FPGA Typen 24 sog. GTH Transceiver zur Verfügung, welche serielle Datenströme mit 10 GBit/s verarbeiten und insbesondere für eine möglichst einfache Integration von 10G Ethernet Schnittstellen konzipiert wurden: Zur Bereitstellung einer optischen 10GBase-SR Schnittstelle werden neben einem passenden Small Form-factor Pluggable

Plus (SFP+) Transceiver nur noch wenige passive elektrische Bauteile benötigt. Sämtliche für die physikalische und Netzwerkschicht benötigte Logik lässt sich mit Hilfe eines GTH Transceivers und ein wenig Logik im FPGA realisieren. Diese einfache Integrationsmöglichkeit von 10G Ethernet ist ein weiterer Pluspunkt der für die Verwendung der beiden oben genannten FPGA Typen spricht: Eine sinnvolle Integration des Kommunikationssystems in eine bestehende Infrastruktur bedarf selbstverständlich auch einer schnellen Datenverbindung der Vermittlungsstelle zu anderen Zugangs- sowie Weitverkehrsnetzen. Dabei kommt 10G Ethernet besonders häufig zum Einsatz, so dass auch hier eine hohe Kompatibilität zu bestehenden Systemen gegeben ist. Bei einer anvisierten Gesamtbandbreite von 50 GBit/s reichen theoretisch fünf 10G Verbindungen für eine Anbindung aus, in der Praxis wird man eher sechs oder sieben Links vorsehen, um Protokolloverhead zu kompensieren und Latenzen zu verringern. Die restlichen am FPGA verfügbaren GTH Schnittstellen können auf ähnlich einfache Weise genutzt werden, um mehrere FPGAs miteinander zu verbinden, beispielsweise zur Aufteilung der Signalprozessierung auf mehrere Bausteine.

Neben der verfügbaren Kommunikationsbandbreite bezüglich der DACs und ADCs bzw. weiteren Netzwerken und FPGAs, darf auch die Anbindung von schnellem Speicher an den FPGA nicht unterschätzt werden. Dieser wird vor allem benötigt, um Datenpakete umzusortieren falls Paketströme vieler Teilnehmer mit unterschiedlichen Prioritäten und verschiedenen Quality-of-Service (QoS) Anforderungen gemeinsam verarbeitet werden. Dieser Fall tritt beispielsweise auf, wenn Internet Protocol (IP)-Telefonie oder IP-TV parallel zu normalem Internetverkehr am selben Teilnehmeranschluss verwendet wird, die verschiedenen Dienste jedoch an unterschiedlichen 10G Ports der Vermittlungsstelle bereitgestellt werden. Der zum Planungszeitpunkt des Systems leistungsfähigste Speicher, welcher auch mit den oben genannten FPGA Typen kompatibel ist, ist vom Typ Double Data Rate 3 (DDR3). Bei der Virtex-6 FPGA Serie werden normale IO Pins zur Anbindung von Speicher verwendet. Aufgrund der hohen Anforderungen an Timing und Signalqualität von DDR3 Speicher existieren jedoch zahlreiche Einschränkungen bzgl. der möglichen Signalverbindungen zwischen Speichermodul und FPGA. Nach einer Abschätzung über zusätzlich benötigte Schnittstellen (Universal Asynchronous Receiver Transmitter (UART), 1G Ethernet, FPGA Mezzanine

Card (FMC)-Port, ...) konnte mit Hilfe des Xilinx Memory Wizards die maximal mögliche Speicherbandbreite ermittelt werden: Es können maximal drei 64-Bit Small Outline Dual Inline Memory Module (SO-DIMM) DDR3-1066 Module, wie sie auch in gängigen Laptops zum Einsatz kommen, verbunden werden. Bei einer Betriebsfrequenz von 533 MHz und insgesamt 192 parallel genutzten Datenleitungen ist somit ein Durchsatz von ca. 25 GByte/s möglich. Dieser Wert passt sehr gut zur anvisierten Gesamtbandbreite des geplanten Zugangsnetzes von 50 GBit/s: Für den Fall, dass jedes Datenpaket einmal in den Speicher geschrieben und einmal von ihm gelesen wird, liegt die benötigte Speicherbandbreite pro Übertragungsrichtung bei 12.5 GByte/s und entsprechend bei 25 GByte/s für einen voll ausgelasteten Full-duplex Betrieb des Systems. Weitere Details zur Anbindung des Speichers finden sich in Kapitel 6.1.

Zusammenfassend kann festgehalten werden, dass die beiden eingangs genannten FPGA Typen zumindest bzgl. ihrer IO Bandbreite für den Einsatz in der geplanten Vermittlungsstelle geeignet sind. Aufgrund der Limitierung von Konkurrenzprodukten bei der verfügbaren Anzahl von MGTs im 6.25 GBit/s Bereich kommen jedoch auch keine weiteren Produkte in Frage. Zumindest in diesem Punkt kann also durchaus davon gesprochen werden, die Grenzen des technisch Machbaren auszuloten. Was die verfügbaren Logikressourcen betrifft, so bewegen sich die genannten FPGA Typen ebenfalls im oberen Bereich aktuell verfügbarer Technologien. Durch die Möglichkeit mehrere FPGA Boards mit Hilfe von GTH Links hoch performant miteinander zu koppeln, spielt dieser Punkt im Vergleich zur verfügbaren IO Bandbreite jedoch auch eine untergeordnetere Rolle.

Wie bereits zu Beginn dieses Abschnitts erwähnt, erfordert die Vermittlungsstelle aufgrund der aggregierten Prozessierung der gesamten Netzwerkbandbreite die höchste Signalverarbeitungsleistung sowie die schnellsten DACs und ADCs. Aufgrund der Leistungsgrenzen verfügbarer Komponenten sind damit die Systemparameter weitgehend festgelegt. Trotzdem soll hier der Vollständigkeit halber noch kurz auf die Anforderungen der Teilnehmeranschlüsse eingegangen werden. Wie auch die Vermittlungsstelle, muss ein Teilnehmeranschluss zumindest über einen 10G Ethernet Port für die Nutzdaten verfügen, da die insgesamt erzielbare Datenrate im Bereich einiger GBit/s liegen wird. Die Abstraten der benötigten AD und DA Wandler liegen hingegen deutlich niedriger. Die

hier verwendeten 3.125 GSa/s bilden einen guten Kompromiss zwischen den Kosten für die Wandler und deren Integration in das geplante Zugangsnetzwerk. Im Gegensatz zur Vermittlungsstelle kommen Wandler mit einer Auflösung von zwölf Bit und damit deutlich besserem Dynamikbereich zum Einsatz. Das erlaubt eine leistungsfähigere Realisierung von Digital Signal Processing (DSP) basierten Signaloptimierungen auf der Teilnehmerseite, um die schlechte Wandlerauflösung der Vermittlungsstelle zumindest teilweise zu kompensieren. Ein Nachteil ist, dass diese zwölf Bit Wandler immerhin noch ein Viertel der IO-Bandbreite der Wandler an der Vermittlungsstelle benötigen. Da auch an einem Teilnehmeranschluss eine gewisse Menge Speicher am FPGA verfügbar sein muss, werden die AD und DA Wandler auch hier mit Hilfe der GTX Transceiver mit dem FPGA verbunden. Somit kann im Experiment weitgehend das gleiche Board für die Vermittlungsstelle sowie die Teilnehmeranschlüsse verwendet werden, wodurch wiederum Kosten eingespart werden.

### 4.1.3.2. Komponentenintegration

Die Verwendung der eingangs genannten FPGA Typen birgt jedoch einen großen Nachteil: Es ist kein Entwicklungsboard verfügbar, welches auf den oben genannten FPGAs im benötigten Gehäuse basiert, so dass alle 48 GTX Transceiver zur Verfügung stehen. Die einzige Möglichkeit zur Verwendung dieser FPGAs besteht daher in der Entwicklung einer eigenen und speziell angefertigten Platine. Dies erlaubt einerseits natürlich auch gleich die Integration passender Sockel zur Aufnahme der DACs, ADCs, SFP+ Transceiver und Speichermodule, so dass viele Adapterplatinen und Kabel eingespart werden können. Andererseits handelt es sich bei den benötigten Signalfrequenzen und Bandbreiten um ein sehr komplexes und kostenintensives Design, welches aufgrund des benötigten Knowhows im Layout sowie der Komplexität von Platinenproduktion und Bestückung auch extern ausgeführt werden muss. Glücklicherweise ermöglichten sehr gute Beziehungen zu Xilinx den weitgehend kostenlosen Bezug gleich einiger der benötigten FPGA Bausteine. Somit war ein sehr großer Kostenpunkt bereits abgedeckt. Bedenkt man zusätzlich, dass auch kommerzielle Entwicklungsboards in dieser Leistungsklasse schnell im fünfstelligen Bereich liegen (vgl. [1]) und für eine Demons-

tration des gesamten Kommunikationssystems gleich mehrere solcher Platinen benötigt werden, relativieren sich die Kosten für ein eigenes Design.

Aus diesem Grund wurde entschieden, ein eigenes, für das System passendes Board basierend auf einem Xilinx Virtex-6 FPGA des Typs XC6VHX565T-2FFG1924 zu entwickeln und zu produzieren (vgl. Kapitel 6.1). Es soll für die Vermittlungsstelle sowie für die Teilnehmeranschlüsse verwendet werden und muss daher universell ausgelegt sein. Da bei den Teilnehmern generell weniger IO-Bandbreite für die AD und DA Wandler und somit auch weniger Logikressourcen benötigt werden, ist ein für die Vermittlungsstelle entwickeltes Board auch hier absolut ausreichend. Der Hauptunterschied besteht in den verwendeten DA und AD Wandlern. Mit einer Abtastrate von 3.125 GSa/s bei einer Auflösung von 12 Bit benötigen sie zwar deutlich weniger IO-Bandbreite als die Micram Wandler, es handelt sich jedoch immer noch um einen relativ hohen Wert. Die gewählte Lösung besteht darin, Adapterplatinen auf die Sockel der Micram Wandler zu setzen, welche die GTX Links mit den Wandlern der Teilnehmeranschlüsse verbinden. Mehr dazu in Kapitel 6.3.

### 4.1.4. Aufgabenverteilung und Netzsynchronisation

An der Vermittlungsstelle, welche das gesamte Datenaufkommen des Zugangsnetzwerks parallel verarbeiten muss, fallen enorme Mengen zu prozessierender Daten an. Bedenkt man, dass typische FPGA Designs selbst bei modernen Virtex-6 Typen lediglich mit einigen 100 MHz takten, erkennt man, wie viele Samples beim Anschluss zweier DA bzw. AD Wandler mit einer Abtastrate von 25 GSa/s parallel bearbeitet werden müssen. Zur Reduktion von Ressourcenverbrauch und Verdrahtungskomplexität wurde in dieser Arbeit eine Taktfrequenz von 390,625 MHz für die OFDM Prozessierung im FPGA gewählt, dies entspricht einem 64-tel der Wandler Abtastrate.

Die aufwändigste Operation in einem OFDM Empfänger bzw. Sender ist typischerweise die Berechnung der FFT bzw. deren Inversen, der (IFFT). Wie in Kapitel 5.2 im Detail erläutert wird, benötigt bereits eine einzelne 256-Punkte FFT bei dem benötigten Parallelisierungsgrad von 64 (komplexen) Samples pro Takt ca. 40% der verfügbaren FPGA

Ressourcen. Da neben der FFT noch weitere rechenintensive Schritte im FPGA der Vermittlungsstelle ausgeführt werden müssen, ist eine parallele Implementierung des OFDM Senders und Empfängers in einem FPGA an der Vermittlungsstelle nicht möglich. Vielmehr werden dazu zwei der gefertigten FPGA-Platinen benötigt, wobei jeweils eine die OFDM Prozessierung der Sendedaten durchführt und die andere die Empfangsdaten verarbeitet. Zusätzlich werden, sofern möglich, weitere Schritte der OFDM-Signalverarbeitungskette auf die Teilnehmeranschlüsse ausgelagert. Das betrifft vor allem eine Vorverarbeitung der Sendedaten beim Teilnehmer für den Uplink. Hier wird bereits im Sender des Teilnehmeranschlusses Phasenlage und Amplitude der einzelnen OFDM Subträger korrigiert, so dass diese mit optimaler Leistung und minimalem Phasenversatz am Empfänger der Vermittlungsstation eintreffen. Der Uplink-Empfänger der Vermittlungsstelle benötigt somit nur noch eine verhältnismäßig einfache Schaltung zur Erkennung spezieller Testsignale, welche zudem nicht in Echtzeit erfolgen muss. Die ermittelten Korrekturdaten können mit Hilfe des Downlinks dem entsprechenden Teilnehmer mitgeteilt werden, so dass dieser die entsprechend benötigte Vorverarbeitung der Subträger durchführen kann. Diese so genannte Pre-Equalization ist so einfach möglich, da es sich bei dem verwendeten Übertragungsmedium (eine Glasfaser) um einen Kanal handelt, dessen Übertragungsparameter sich nur relativ langsam, z.B. durch Temperaturvariation, ändern und sämtliche Oszillatoren der Teilnehmer auf die Master-Oszillatoren an der Vermittlungsstelle gelockt werden.

Aus diesen Randbedingungen ergibt sich direkt die Abfolge der Netzsynchrosation: Zunächst wird eine Downlink Verbindung von der Vermittlungsstelle zum Teilnehmer aufgebaut. Mithilfe dieser unidirektionalen Verbindung ist die Vermittlungsstelle nun in der Lage, Kommandos zu den Teilnehmern zu senden. Die Teilnehmer werden der Reihe nach angewiesen, entsprechende Testsignale zur Vermittlungsstelle zu senden, woraus Korrekturdaten für den Pre-Equalizer der Teilnehmer sowie *Timing-Advance* Parameter gewonnen werden. Dabei handelt es sich um einen Wert der angibt, zu welchem Zeitpunkt in Relation zum Rahmenbeginn des empfangenden Downlinks ein Teilnehmer mit der Übertragung seines Uplinkrahmens beginnen muss, so dass sich die Rahmen aller Teilnehmer im Netzwerk an der Vermittlungsstelle exakt überlagern. Typischerweise ist dieser Wert primär von der Entfernung des Teilnehmers

zur Vermittlungsstelle abhängig. Mittels eines zweiten Kommandos werden die ermittelten Korrekturwerte zu den Teilnehmern gesendet. Nun ist auch der Uplink einsatzbereit.

### 4.1.5. Bandbreitenverteilung

In einem OFDM System werden typischerweise nicht alle Subträger zur Datenübermittlung verwendet. Subträger, deren Amplitude während der Übertragung zu Gleichspannungsanteilen führt (sog. Direct Current (DC) Träger), werden überhaupt nicht moduliert. Weitere Subträger werden mit konstanter oder in der Abfolge klar definierter Amplitude und Phase moduliert. Diese sogenannten Pilottöne tragen ebenfalls keinerlei Information. Sie werden vom Empfänger ausgewertet und erlauben diesem so mittels Interpolation die Amplitude und die Phasenlage der Daten-subträger des OFDM Systems zu ermitteln. Die Anzahl der verwendeten Pilottöne variiert je nach Realisierung des OFDM Systems. Ein Minimum von zwei Pilottönen ist ausreichend um Frequenzfehler von Lokal- und Samplingoszillatoren zu erkennen. Ist die Übertragungstrecke sehr anfällig für Störungen und Phasenverschiebungen, wird typischerweise eine größere Anzahl von Pilotträgern verwendet.

Das in dieser Arbeit präsentierte Kommunikationssystem basiert auf einer optischen Übertragungstrecke. Es kann daher davon ausgegangen werden, dass die Übertragungstrecke relativ störunanfällig ist und auch Phasenverschiebungen des Übertragungssignals nur relativ langsam auftreten. Es muss jedoch beachtet werden, dass jeder Teilnehmer nur einen begrenzten Bereich der Gesamtbandbreite empfängt, weshalb trotzdem insgesamt verhältnismäßig viele Pilottöne eingesetzt werden müssen. Zur sicheren Erkennung von Frequenzfehlern werden daher zwei Pilottöne pro Teilnehmer eingesetzt. Ein weiterer Subträger pro Teilnehmer kann aufgrund von Gleichspannungsanteilen nicht moduliert werden. Somit ergeben sich bei einer Gesamtzahl von 16 Subträgern pro Teilnehmer 13 zur Datenübertragung nutzbare Subträger.

Eine weitere Reduktion der zur Datenübertragung verfügbaren Bandbreite erfolgt durch die Verwendung des sogenannten Cyclic Prefix (CP). Dabei handelt es sich um einen kleinen zeitlichen Ausschnitt des erzeugten OFDM Symbols, welcher dem Symbol vorangestellt wird und



somit die Gesamtsymboldauer um einen gewissen Anteil erhöht. Dieser CP wird benötigt, um Dispersionseffekte des Übertragungsmediums auszugleichen und der auf Autokorrelation basierenden Symbolgrenzerkennung im Empfänger einen gewissen Spielraum zu verschaffen. Typische Werte für die Länge eines CP liegen im Bereich von einem Sechzehntel bis zu einem Viertel der OFDM Symbollänge, wobei aufgrund von Vorteilen in der FPGA Realisierung in dieser Arbeit eine Länge von einem Viertel verwendet wird.

Werden noch weitere Effekte wie beispielsweise Protokolloverhead hinzugechnet, fällt auf, dass bei dem in dieser Arbeit vorgestellten Kommunikationssystem die Nutzbandbreite nur gut die Hälfte der theoretisch verfügbaren Bandbreite ausmacht. Durch eine Optimierung der einzelnen Verarbeitungsschritte ließe sich dieser Wert sicherlich noch deutlich steigern, jedoch sollte beachtet werden, dass bei komplexen Modulationsverfahren ein nicht unerheblicher Anteil der verfügbaren Bandbreite vom System selbst beansprucht wird.

Im nächsten Teil dieses Kapitels erfolgt nun eine detaillierte Beschreibung der gewählten Systemparameter und definierten Übertragungsprotokolle. Anschließend wird die Hardwarearchitektur der eingesetzten Signalverarbeitung beschrieben.

## 4.2. Spezifikation der Systemparameter und Kommunikationsprotokolle

### 4.2.1. Grundlegende Systemparameter

Einige grundlegende Systemparameter wurden bereits im vorigen Abschnitt dieses Kapitels festgelegt. Sie ergeben sich aus grundlegenden Überlegungen zu den an das Kommunikationssystem gestellten Anforderungen und der eingesetzten Hardware. Insbesondere betrifft dies die gewählten Abtastraten und eine sinnvolle Wahl der Größe für die verwendeten FFT Algorithmen. Sie werden in Tabelle 4.1 noch einmal zusammengefasst.

## 4. Systemarchitektur

---

Parameter	Wert
ONU Samplerate DAC & ADC	3.125 GSa/s
OLT Samplerate DAC & ADC	25 GSa/s
OLT FFT Größe	256
OLT Cyclic prefix	25%
OLT Gesamtdatendurchsatz (pro Übertragungsrichtung)	50 GBit/s
OLT Gesamtbandbreite	25 GHz

Tabelle 4.1.: OFDM Systemparameter

Basierend auf diesen bereits festgelegten Parametern können nun alle weiteren definiert werden. Dazu erfolgt im nächsten Abschnitt zunächst eine Betrachtung des OFDM Symbolaufbaus im Frequenzbereich und anschließend im Zeitbereich.

### 4.2.2. OFDM Signalaufbau im Frequenzbereich

Die Gesamtanzahl der Subträger am OLT ist durch die eingesetzte FFT Größe auf 256 festgelegt. Bei einer Gesamtnetzbandbreite von 25 GHz ergibt sich somit ein Subträgerabstand von 97.65625 MHz. Ein wesentliches Merkmal der in dieser Arbeit eingesetzten OFDMA Architektur ist die Unterteilung des Gesamtspektrums in einzelne Teilspektren, die von den Teilnehmern verarbeitet werden können. Da für eine korrekte Funktion des Systems der Subträgerabstand auch am Teilnehmer 97.65625 MHz betragen muss, ergibt sich ein Maximum von 32 Subträgern, die ein Teilnehmer mit seinen bei 3.125 GSa/s arbeitenden AD bzw. DA Wandlern verarbeiten kann.

In der Praxis wäre eine Konfiguration mit 32 genutzten Subträgern pro Teilnehmer jedoch nur äußerst schwierig realisierbar. Die analogen Antialiasing Filter der AD bzw. DA Wandler in den Teilnehmern müssten extrem steilflankig ausgelegt sein, um einerseits die Subträger der benachbarten Bereiche im vom OLT gesendeten Gesamtspektrum sauber zu unterdrücken und andererseits die am Rande des gewählten Bereichs befindlichen Subträger möglichst unverfälscht passieren zu lassen. Aus diesem Grund wurde in dem hier vorgestellten System ein anderer Weg

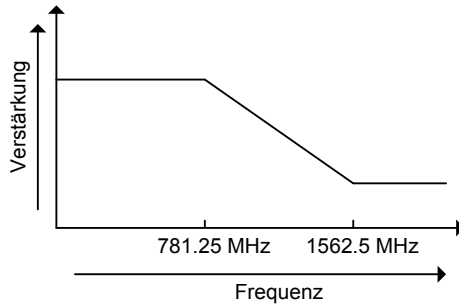


Abbildung 4.3.: Analoges Tiefpassfilter am ONU

gewählt. Anstelle von 32 Subträgern werden lediglich die niederfrequenten 16 Subträger von einem Teilnehmer ausgewertet. Entsprechend einfach können die eingesetzten analogen Tiefpassfilter konstruiert werden, da nun der Übergangsbereich zwischen Durchlass- und Stopband der Filter genauso breit ist, wie der eigentliche Durchlassbereich (siehe Abbildung 4.3). Die Anzahl der OFDMA-Subträger, die von einem Teilnehmer simultan verarbeitet werden, beträgt also 16. Entsprechend lassen sich auch die 256 Subträger des Gesamtspektrums in 16 Gruppen unterteilen, die im Folgenden als *spektrale Gruppen* bezeichnet werden.

Auch die Antialiasing Filter an den AD bzw. DA Wandlern der Vermittlungsstelle können nicht beliebig steiflankig ausgelegt werden. Daher können auch im Randbereich des Gesamtspektrums nicht alle Subträger eingesetzt werden, so dass die beiden äußersten spektralen Gruppen ungenutzt bleiben müssen. Abbildung 4.4 zeigt den Aufbau des Gesamtspektrums. Dabei sollte beachtet werden, dass die physikalische Indizierung der Subträger von -128 bis +127 angegeben ist (vgl. Kapitel 2.1.3). Sie entspricht dem Frequenzoffset des Subträgers bezüglich der Frequenz eines Trägersignals, welches mit Hilfe des IQ-Basisbandsignals moduliert wurde. Der Index gibt dabei den Frequenzabstand in Vielfachen des Subträgerabstands von 97.65625 MHz an. Alternativ zum physikalischen Index existiert auch ein logischer Index, welcher der Nummerierung der Punkte bei der Implementierung einer FFT entspricht. Zusätzlich wird hier noch zwischen einem globalen und einem lokalen Index unterschieden. Der globale Index bezieht sich auf die Mittenfrequenz des

## 4. Systemarchitektur

Gesamtspektrums, welches von der Vermittlungsstelle verarbeitet wird und kommt entsprechend bei der Subträgerindizierung am OLT zum Einsatz. Der lokale Index beschreibt hingegen einen Subträger bezogen auf die Mittenfrequenz der gerade gewählten spektralen Gruppe eines Teilnehmers und wird folglich primär bei der Verarbeitung im ONU genutzt.

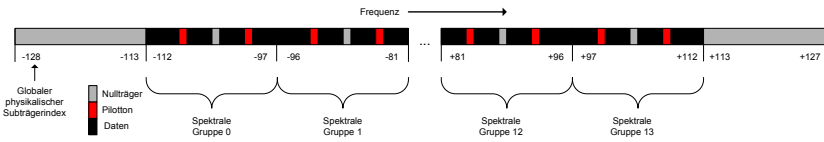


Abbildung 4.4.: Spektraler Aufbau des OFDM Signals

Wie bereits erwähnt, muss der mittlere Subträger in jeder spektralen Gruppe unmoduliert bleiben, um Probleme mit DC Offsets zu vermeiden. Weiterhin werden mindestens zwei Pilottöne pro Gruppe benötigt um Frequenzfehler der Oszillatoren der Teilnehmer auszugleichen. Glücklicherweise handelt es sich bei einer Glasfaser jedoch um einen sehr hochwertigen Übertragungskanal, sodass nicht mit Fading- und starken Dispersionseffekten zu rechnen ist und zwei Pilottöne in diesem Fall völlig ausreichend sind. Die Platzierung der Pilottöne in der Gruppe erfolgt bei den lokalen physikalischen Subträgerindizes +4 und -4. Sie befinden sich somit ausreichend weit voneinander entfernt, um Frequenzfehler zuverlässig zu erkennen, haben jedoch ebenfalls noch einen gewissen Abstand zu den Grenzen der spektralen Gruppe, sodass sie von den analogen Filtern möglichst wenig beeinflusst werden.

Die restlichen 13 Subträger einer spektralen Gruppe stehen damit zur Datenübertragung zur Verfügung. Zur Wahrung der Symmetrie im Gesamtspektrum und zur Vermeidung eines modulierten globalen DC Subträgers (globaler Index 0) werden in den niederfrequenten Gruppen die lokalen physikalischen Indizes -8 bis +7 für den Datentransfer verwendet, während in den höherfrequenten Gruppen die Indizes -7 bis +8 zum Einsatz kommen. Für den Zugang zu einer dieser 14 spektralen Gruppen muss ein Teilnehmer seinen Lokaloszillator auf die Mittenfrequenz der gewünschten Gruppe abstimmen und erhält somit Zugriff auf die beiden Pilottöne sowie die 13 Datensubträger. Abbildung 4.5 fasst

alle Angaben in Form eines detaillierten Subträgerbelegungsplans des Kommunikationssystems noch einmal zusammen.

### 4.2.3. OFDM Signalaufbau im Zeitbereich

Im Zeitbereich erfolgt die Übertragung in Frames mit einer festen Länge. Jeder Frame ist in vier Bereiche mit unterschiedlichen Aufgaben unterteilt, wobei die zeitliche Grundeinheit durch ein OFDM Symbol gebildet wird. Weiterhin gestaltet sich der Aufbau der Frames im Uplink und Downlink grundsätzlich identisch, insbesondere ihre Länge und Einteilung in vier Bereiche stimmen überein, wodurch die Synchronisation der Up- und Downlinksignale stark vereinfacht wird. Im Folgenden wird zunächst der Aufbau eines Frames für den Downlink beschrieben. Der Uplink unterscheidet sich vor allem dadurch, dass die Synchronisationsinformationen in den ersten beiden Bereichen eines Frames nur nach Anfrage vom OLT von einer ONU gesendet werden.

Zu Beginn eines jeden Frames erfolgt die Übermittlung einer sogenannten Präambel, auch Sync-Sequenz genannt. Sie enthält zehn exakt festgelegte Symbole, welche zur Zeit- und Frequenzsynchronisation verwendet werden. Mit Hilfe einer Autokorrelation wird diese Sequenz am Empfänger erkannt und ermöglicht aufgrund einer geschickten Wahl der verwendeten Symbole die Korrektur der Frequenzoffsets von Lokal- und Samplingoszillatoren. Außerdem wird so der exakte Startzeitpunkt eines Frames erkannt.

Der zweite Bereich des Frames besteht aus 16 Symbolen und dient der Ermittlung der genauen Phasenreferenz jedes einzelnen Subträgers. Dazu werden zunächst die beiden Pilottöne jeder spektralen Gruppe mit dem konstanten Wert  $1+0j$  moduliert und anschließend der Reihe nach die einzelnen Datensubträger. Abgesehen von den beiden Pilottönen, welche außer während der Sync-Sequenz in jedem OFDM Symbol aktiv sind, wird während der Übermittlung der Phasenreferenz jeweils nur ein einzelner Datenträger pro spektraler Gruppe moduliert. Der Grund dafür liegt in der geringen DAC Auflösung des OLT von nur 6 Bit: Aufgrund des relativ hohen PAPR von OFDM wird das Sendesignal am OLT stark begrenzt, so dass bei einer simultanen Modulation aller Subträger das Ausgangssignal stark geclippt würde. Dies hätte einen negativen Ein-

# 4. Systemarchitektur

Special Carriers	Global Logical Index	Global Physical Index	Spectral Group	Local Physical Index	Local Logical Index	Special Carriers	Special Carriers	Global Logical Index	Global Physical Index	Spectral Group	Local Physical Index	Local Logical Index	Special Carriers
Zero	128	128	Guardband				Zero	0	0	Spectral Group 7 (+35.25 MHz)			Zero
Zero	130	129					Zero	2	-2		-6	-28	Zero
Zero	131	130					Zero	3	-3		-7	-27	Zero
Zero	132	131					Zero	4	-4		-8	-26	Zero
Zero	133	132					Zero	5	-5		-9	-25	Zero
Zero	134	133					Zero	6	-6		-10	-24	Zero
Zero	135	134					Zero	7	-7		-11	-23	Zero
Zero	136	135					Zero	8	-8		-12	-22	Zero
Zero	137	136					Zero	9	-9		-13	-21	Zero
Zero	138	137					Zero	10	-10		-14	-20	Zero
Zero	139	138					Zero	11	-11		-15	-19	Zero
Zero	140	139					Zero	12	-12		-16	-18	Zero
Zero	141	140					Zero	13	-13		-17	-17	Zero
Zero	142	141					Zero	14	-14		-18	-16	Zero
Zero	143	142					Zero	15	-15		-19	-15	Zero
Zero	144	143					Zero	16	-16		-20	-14	Zero
Zero	145	144					Zero	17	-17		-21	-13	Zero
Zero	146	145					Zero	18	-18		-22	-12	Zero
Zero	147	146					Zero	19	-19		-23	-11	Zero
Zero	148	147					Zero	20	-20		-24	-10	Zero
Zero	149	148				Zero	21	-21	-25	-9	Zero		
Zero	150	149				Zero	22	-22	-26	-8	Zero		
Zero	151	150				Zero	23	-23	-27	-7	Zero		
Zero	152	151				Zero	24	-24	-28	-6	Zero		
Zero	153	152				Zero	25	-25	-29	-5	Zero		
Zero	154	153				Zero	26	-26	-30	-4	Zero		
Zero	155	154				Zero	27	-27	-31	-3	Zero		
Zero	156	155				Zero	28	-28	-32	-2	Zero		
Zero	157	156				Zero	29	-29	-33	-1	Zero		
Zero	158	157				Zero	30	-30	-34	0	Zero		
Zero	159	158				Zero	31	-31	-35	1	Zero		
Zero	160	159				Zero	32	-32	-36	2	Zero		
Zero	161	160				Zero	33	-33	-37	3	Zero		
Zero	162	161				Zero	34	-34	-38	4	Zero		
Zero	163	162				Zero	35	-35	-39	5	Zero		
Zero	164	163				Zero	36	-36	-40	6	Zero		
Zero	165	164				Zero	37	-37	-41	7	Zero		
Zero	166	165				Zero	38	-38	-42	8	Zero		
Zero	167	166				Zero	39	-39	-43	9	Zero		
Zero	168	167				Zero	40	-40	-44	10	Zero		
Zero	169	168				Zero	41	-41	-45	11	Zero		
Zero	170	169				Zero	42	-42	-46	12	Zero		
Zero	171	170				Zero	43	-43	-47	13	Zero		
Zero	172	171				Zero	44	-44	-48	14	Zero		
Zero	173	172				Zero	45	-45	-49	15	Zero		
Zero	174	173				Zero	46	-46	-50	16	Zero		
Zero	175	174				Zero	47	-47	-51	17	Zero		
Zero	176	175				Zero	48	-48	-52	18	Zero		
Zero	177	176				Zero	49	-49	-53	19	Zero		
Zero	178	177				Zero	50	-50	-54	20	Zero		
Zero	179	178				Zero	51	-51	-55	21	Zero		
Zero	180	179				Zero	52	-52	-56	22	Zero		
Zero	181	180				Zero	53	-53	-57	23	Zero		
Zero	182	181				Zero	54	-54	-58	24	Zero		
Zero	183	182				Zero	55	-55	-59	25	Zero		
Zero	184	183				Zero	56	-56	-60	26	Zero		
Zero	185	184				Zero	57	-57	-61	27	Zero		
Zero	186	185				Zero	58	-58	-62	28	Zero		
Zero	187	186				Zero	59	-59	-63	29	Zero		
Zero	188	187				Zero	60	-60	-64	30	Zero		
Zero	189	188				Zero	61	-61	-65	31	Zero		
Zero	190	189				Zero	62	-62	-66	32	Zero		
Zero	191	190				Zero	63	-63	-67	33	Zero		
Zero	192	191				Zero	64	-64	-68	34	Zero		
Zero	193	192				Zero	65	-65	-69	35	Zero		
Zero	194	193				Zero	66	-66	-70	36	Zero		
Zero	195	194				Zero	67	-67	-71	37	Zero		
Zero	196	195				Zero	68	-68	-72	38	Zero		
Zero	197	196				Zero	69	-69	-73	39	Zero		
Zero	198	197				Zero	70	-70	-74	40	Zero		
Zero	199	198				Zero	71	-71	-75	41	Zero		
Zero	200	199				Zero	72	-72	-76	42	Zero		
Zero	201	200				Zero	73	-73	-77	43	Zero		
Zero	202	201				Zero	74	-74	-78	44	Zero		
Zero	203	202				Zero	75	-75	-79	45	Zero		
Zero	204	203				Zero	76	-76	-80	46	Zero		
Zero	205	204				Zero	77	-77	-81	47	Zero		
Zero	206	205				Zero	78	-78	-82	48	Zero		
Zero	207	206				Zero	79	-79	-83	49	Zero		
Zero	208	207				Zero	80	-80	-84	50	Zero		
Zero	209	208				Zero	81	-81	-85	51	Zero		
Zero	210	209				Zero	82	-82	-86	52	Zero		
Zero	211	210				Zero	83	-83	-87	53	Zero		
Zero	212	211				Zero	84	-84	-88	54	Zero		
Zero	213	212				Zero	85	-85	-89	55	Zero		
Zero	214	213				Zero	86	-86	-90	56	Zero		
Zero	215	214				Zero	87	-87	-91	57	Zero		
Zero	216	215				Zero	88	-88	-92	58	Zero		
Zero	217	216				Zero	89	-89	-93	59	Zero		
Zero	218	217				Zero	90	-90	-94	60	Zero		
Zero	219	218				Zero	91	-91	-95	61	Zero		
Zero	220	219				Zero	92	-92	-96	62	Zero		
Zero	221	220				Zero	93	-93	-97	63	Zero		
Zero	222	221				Zero	94	-94	-98	64	Zero		
Zero	223	222				Zero	95	-95	-99	65	Zero		
Zero	224	223				Zero	96	-96	-100	66	Zero		
Zero	225	224				Zero	97	-97	-101	67	Zero		
Zero	226	225				Zero	98	-98	-102	68	Zero		
Zero	227	226				Zero	99	-99	-103	69	Zero		
Zero	228	227				Zero	100	+00	-104	70	Zero		
Zero	229	228				Zero	101	+01	-105	71	Zero		
Zero	230	229				Zero	102	+02	-106	72	Zero		
Zero	231	230				Zero	103	+03	-107	73	Zero		
Zero	232	231				Zero	104	+04	-108	74	Zero		
Zero	233	232				Zero	105	+05	-109	75	Zero		
Zero	234	233				Zero	106	+06	-110	76	Zero		
Zero	235	234				Zero	107	+07	-111	77	Zero		
Zero	236	235				Zero	108	+08	-112	78	Zero		
Zero	237	236				Zero	109	+09	-113	79	Zero		
Zero	238	237				Zero	110	+10	-114	80	Zero		
Zero	239	238				Zero	111	+11	-115	81	Zero		
Zero	240	239				Zero	112	+12	-116	82	Zero		
Zero	241	240				Zero	113	+13	-117	83	Zero		
Zero	242	241				Zero	114	+14	-118	84	Zero		
Zero	243	242				Zero	115	+15	-119	85	Zero		
Zero	244	243				Zero	116	+16	-120	86	Zero		
Zero	245	244				Zero	117	+17	-121	87	Zero		
Zero	246	245				Zero	118	+18	-122	88	Zero		
Zero	247	246				Zero	119	+19	-123	89	Zero		
Zero	248	247				Zero	120	+20	-124	90	Zero		
Zero	249	248				Zero	121	+21	-125	91	Zero		
Zero	250	249				Zero	122	+22	-126	92	Zero		
Zero	251	250				Zero	123	+23	-127	93	Zero		
Zero	252	251				Zero	124	+24	-128	94	Zero		
Zero	253	252				Zero	125	+25	-129	95	Zero		
Zero	254	253				Zero	126	+26	-130	96	Zero		
Zero	255	254				Zero	127	+27	-131	97	Zero		
Zero	256	255				Zero	128	+28	-132	98	Zero		
Zero	257	256				Zero	129	+29	-133	99	Zero		
Zero	258	257				Zero	130	+30	-134	100	Zero		

Abbildung 4.5.: OLT Subträgerbelegungsplan

## 4.2. Spezifikation der Systemparameter und Kommunikationsprotokolle

---

fluss auf die Genauigkeit der vom Empfänger ermittelten Phasenlage der Subträger. Werden während der eigentlichen Datenübertragung alle Subträger simultan verwendet, tritt dieser Effekt natürlich auch auf, jedoch können hier einzelne Bitfehler unter Einbezug der FEC leichter abgefangen werden. Die Phasenlage und Amplitude jedes Subträgers wird vom Empfänger gespeichert und durch den bekannten ursprünglichen Zustand jedes Trägers während seiner Aussendung, können individuelle Korrekturfaktoren berechnet werden. Diese werden im Folgenden für eine Korrektur der einzelnen Subträger genutzt (auch Equalization genannt), so dass diese optimal demoduliert werden können. Tabelle 4.2 fasst die aktiven Subträger während der Phasenreferenz noch einmal zusammen.

Symbolindex	Aktive Subträger
0	-4, +4
1	-4, +4
2	-4, +4, -7
3	-4, +4, -6
4	-4, +4, -5
5	-4, +4, -3
6	-4, +4, -2
7	-4, +4, -1
8	-4, +4, +1
9	-4, +4, +2
10	-4, +4, +3
11	-4, +4, +5
12	-4, +4, +6
13	-4, +4, +7
14	-4, +4, +/-8
15	-4, +4

Tabelle 4.2.: Aktive lokale Subträger während einer Phasenreferenz im Downlink

Nach dem Empfang dieses zweiten Teils des Frames kennt der Empfänger sämtliche ausbreitungsbedingten Einflüsse auf das Empfangssignal,

## 4. Systemarchitektur

die er für eine korrekte Dekodierung der in Phase und Amplitude der Subträger kodierten Informationen benötigt. Im dritten Teil des Frames kann daher die Übermittlung von Kontrolldaten erfolgen, welche der Steuerung des Gesamtnetzwerks dienen und in 32 OFDM Symbolen transportiert werden.

Im letzten und längsten Teil des Frames erfolgt schließlich die Übertragung der eigentlichen Nutzdaten. Dazu stehen 8192 Symbole zur Verfügung, wobei jeder Subträger mit einem individuellen Modulationsformat betrieben werden kann. Anschließend an diesen vierten Frameabschnitt folgt sofort der Beginn des nächsten Frames mit der Sync-Sequenz. Eine grafische Darstellung des Frameaufbaus ist in Abbildung 4.6 gegeben.

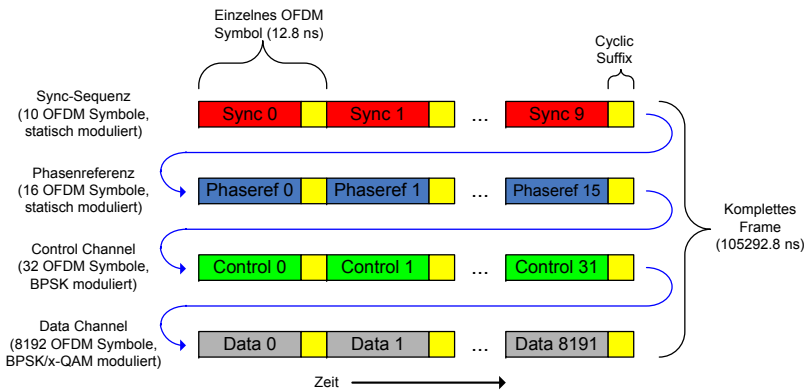


Abbildung 4.6.: Zeitlicher Frameaufbau

Die bisher gemachten Angaben beziehen sich auf den zeitlichen Signalaufbau des Downlinks. Der Uplink ist prinzipiell identisch aufgebaut. Ein wesentlicher Unterschied ist jedoch, dass das OLT im Downlink sämtliche aktiven Subträger des Gesamtnetzwerks gemeinsam kontrolliert, während sich der Uplink aus den von den einzelnen ONUs gesendeten Signalen zusammensetzt, wobei eine ONU nur in der gerade aktiven spektralen Gruppe sendet.



### 4.2.4. Synchronisation

Das Signal des Downlinks wird am OLT gemeinsam für alle spektralen Gruppen generiert. Die Empfänger in den einzelnen ONUs sind dafür verantwortlich, einen lokalen Frametimer mit dem Empfang der Sync-Sequenz des Downlinks zu synchronisieren, sowie die Frequenzen ihrer Lokal- und Samplingoszillatoren zu korrigieren. Dies passiert initial durch die Auswertung der Sync-Sequenz. Da die Framelänge bekannt und fest ist, kann die Frequenzkorrektur des Samplingoszillators einfach durch Messung des zeitlichen Abstands zwischen zwei empfangenen Sync-Sequenzen bestimmt werden. Die Messung des Frequenzoffsets des Lokaloszillators ist nicht so einfach ersichtlich. Zu seiner exakten Bestimmung wird die Sync-Sequenz in zwei identische Bereiche mit einer Länge von jeweils fünf OFDM Symbolen aufgeteilt. Am Empfänger wird nun die Phasenlage der Samples aus dem ersten Bereich mit den Samples des zweiten Bereichs verglichen. Verfügt der Lokaloszillator über einen Frequenzoffset, so ergibt sich ein Phasenoffset zwischen den eigentlich identischen Samples, welcher sich mit Hilfe des bekannten zeitlichen Abstands zwischen den verglichenen Samples in einen Frequenzoffset umrechnen lässt. Siehe hierzu auch die Erläuterungen in Abschnitt 2.2.3.

Mit Hilfe der aus der Sync-Sequenz gewonnenen Korrekturwerte lassen sich die ONU Oszillatoren soweit abstimmen, dass der initiale Empfang eines Frames problemlos möglich ist. Um einen stabilen Empfang des Downlinks auch auf längere Zeit zu ermöglichen, müssen die Oszillatoren am ONU phasenstarr mit ihren Gegenstücken am OLT gekoppelt werden. Dazu werden nach der initialen Synchronisierung die Pilottöne verwendet, indem die Entwicklung ihrer Phase über die Zeit betrachtet wird. Bewegt sich die Phase beider Töne in die selbe Richtung, handelt es sich um einen Common Phase Error (CPE), der eine Phasenverschiebung zwischen den beiden Lokaloszillatoren an OLT und ONU anzeigt. Entwickeln sich die Phasen der Töne in unterschiedliche Richtungen, dann spricht man von einem Differential Phase Error (DPE), der eine Phasenverschiebung zwischen den beiden Samplingoszillatoren an OLT und ONU bedeutet. Der Grund dafür ist, dass sich bei einem Frequenzoffset des Samplingoszillators das vom Empfänger ausgewertete FFT Fenster bzgl. des eintreffenden OFDM Symbols verschiebt, wodurch sich die Phasenlagen von Subträgern mit positiven und negativen physikalischen Indizes

#### 4. Systemarchitektur

---

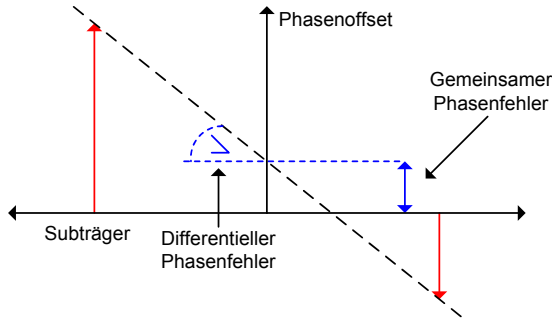


Abbildung 4.7.: Gemeinsamer und Differentieller Phasenfehler

entgegengesetzt bewegen. Beide Verschiebungen können natürlich auch gemeinsam auftreten. Dieser Zusammenhang ist auch in Abbildung 4.7 noch einmal dargestellt.

Zur Korrektur der beiden ONU Oszillatoren wird der CPE bzw. DPE über einen PID Regler auf eine Abweichung von null zurückgeführt. Somit ergibt sich ein geschlossener Regelkreis, der die Lokal- und Sampleoszillatoren am ONU ähnlich einer typischen PLL an die Referenzfrequenzen des OLT koppelt.

Für den Uplink werden sowohl an OLT als auch an ONU die selben Oszillatoren verwendet. Das hat den Vorteil, dass sämtliche Frequenzkorrekturen des Downlinks auch direkt auf den Uplink wirken und der Empfänger am OLT keinerlei Frequenzfehler korrigieren muss. Da die Bandbreite des Uplinks am OLT um ein vielfaches größer als an den ONUs ist, erleichtert dies die Realisierung des Empfängers deutlich. Trotzdem ist auch der Uplink für jede ONU nicht unmittelbar einsatzbereit und muss zunächst synchronisiert werden.

Damit sich am OLT aus den einzelnen Signalanteilen der ONUs auch bei unterschiedlichen Entfernungen zueinander wieder ein vollständiges Spektrum ohne Intersymbolinterferenzen und Frameversatz bilden kann, ist es notwendig, den ONUs individuell mitzuteilen, zu welchem Zeitpunkt sie mit der Übermittlung eines neuen Frames beginnen müssen. Dieses Verfahren ist unter dem Namen *Timing Advance* auch aus anderen Bereichen der digitalen Kommunikation wie z.B. GSM bekannt. Zur Er-

## 4.2. Spezifikation der Systemparameter und Kommunikationsprotokolle

mittlung dieses Parameters übermittelt das OLT der zu konfigurierenden ONU den Befehl, einen Synchronisationsimpuls zu senden. Anschließend übermittelt die angefragte ONU in der Mitte des Sync-Sequenzbereichs des Uplink Frames eine Sinusschwingung mit einer Dauer von exakt zwei Perioden. Eine Periode belegt dabei die Länge eines OFDM Symbols. Gleichzeitig werden am OLT parallel zur Aussendung des Downlinks ebenfalls Uplinkframes aus den empfangenen Daten der AD Wandler gebildet. Eine Kreuzkorrelation jedes empfangenen Symbols mit einer statischen Sinusschwingung erlaubt die Detektion des Synchronisationsimpulses sowie die Bestimmung von dessen Phasenlage. Zusammen mit der Symbolnummer im Uplinkframe am OLT kann somit der korrekte Wert für den *Timing Advance* Parameter errechnet und mit Hilfe des Downlinks zum entsprechenden ONU übertragen werden. Abbildung 4.8 illustriert dieses Vorgehen.

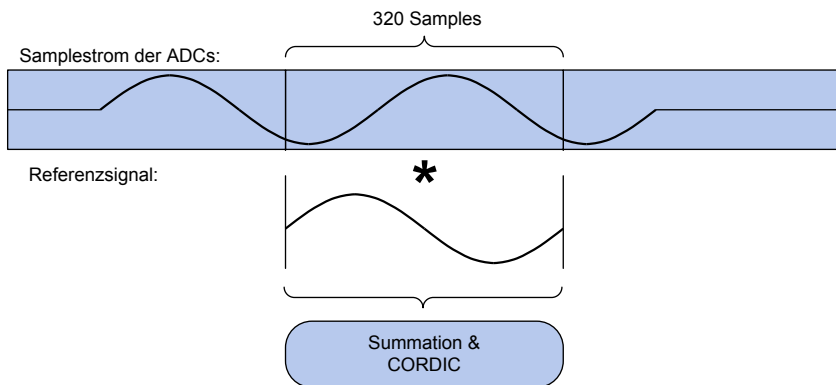


Abbildung 4.8.: Auswertung der Uplink Synchronisationssequenz am OLT

Während des ersten Synchronisationsversuchs ist die Entfernung zwischen OLT und ONU völlig unbekannt, so dass kein sinnvoller *Timing Advance* Wert vorgegeben werden kann. So kann die von einer ONU ausgesendete Synchronisationssequenz zu einem beliebigen Zeitpunkt beim OLT eintreffen und laufende Übertragungen anderer ONUs stören. Aus diesem Grund wird die Datenübertragung aller im Netzwerk registrierter ONUs in Richtung OLT für die Zeitspanne einer Uplinks-

## 4. Systemarchitektur

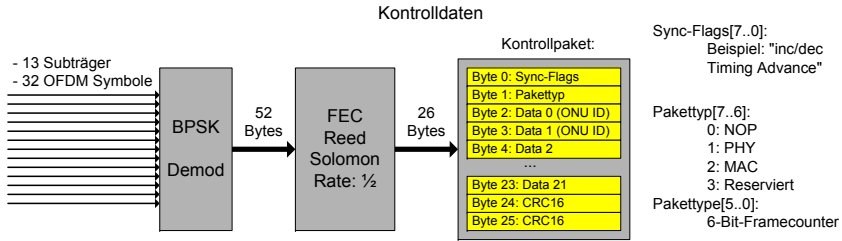


Abbildung 4.9.: Aufbau des Kontrolldatenbereichs

ynchronisation durch entsprechende Kontrollkommandos gestoppt. Da Uplinksynchronisationen nur sehr selten benötigt werden, halten sich deren Auswirkungen auf die Bandbreite des Uplinks in Grenzen.

### 4.2.5. Einbettung von Kontroll- und Nutzdaten

Während die ersten beiden Bereiche eines Frames ausschließlich der Synchronisation des Empfängers auf das eingehende Signal dienen, findet eine tatsächliche Datenübertragung nur in den hinteren beiden Frameteilen statt. Die im dritten Teil übermittelten Kontrolldaten werden unter anderem für die Konfiguration des Uplinks und anderer kritischer Parameter der physikalischen Netzwerkschicht verwendet und sind daher besonders gut gesichert. Abbildung 4.9 zeigt die Einbettung der Kontrolldaten in die übermittelten OFDM Symbole.

Wie leicht zu erkennen ist, werden alle 13 Datensubträger einer spektralen Gruppe parallel zur Übermittlung verwendet. Als Modulationsart kommt mit BPSK ein sehr robustes Verfahren zum Einsatz, welches aber den Nachteil hat, dass pro Datenträger lediglich ein Bit pro Symbol übermittelt wird. Beachtet man jedoch den Verwendungszweck der übermittelten Kontrolldaten wird schnell klar, dass eine sichere Übertragung weitaus wichtiger ist, als ein hoher Durchsatz. Unter Verwendung der 13 Datensubträger einer spektralen Gruppe erreichen ein ONU Empfänger also 13 Bit pro OFDM Symbol. Bei 32 Kontrolldaten-Symbolen ergibt sich eine Gesamtdatenmenge von 416 Bit oder 52 Byte pro Frame. Zur weiteren Absicherung werden die Daten mit Hilfe eines Reed-Solomon-Codes mit

einer Coderate von  $\frac{1}{2}$  geschützt. Diese FEC sichert die Kontrolldaten gegen einzelne Bitfehler und reduziert die nutzbare Paketgröße auf 26 Bytes. Der Aufbau eines Kontrollpaketes ist in Abbildung 4.9 dargestellt: Das erste Byte (*Sync-Flags*) ist für die physikalische Netzwerkschicht reserviert und trägt Kontrollkommandos, etwa um den *Timing Advance* Parameter des Uplink geringfügig zu modifizieren. Das zweite Byte (*Controltype*) bestimmt den Type der Daten im restlichen Paket. Unterschieden wird dabei zwischen Kommandos für die physikalische und MAC Schicht. Ein Beispiel für ein Kommando für die physikalische Schicht ist die Aussendung eines Synchronisationsimpulses im Uplink. Kommandos für die MAC Schicht erlauben hingegen z.B. die Wahl einer neuen spektralen Gruppe. Weiterhin wird in diesem Byte ein Framecounter vom OLT an die ONUs übermittelt. Dieser wird zur Realisierung von Zustandsänderungen im Netzwerk verwendet, die synchron in der Vermittlungsstelle und bei den Teilnehmern ausgeführt werden müssen. Ein Beispiel dafür ist die Änderung des Modulationsverfahrens eines Datenträgers im Nutzdatenbereich. Die folgenden 24 Kontrollbytes enthalten kommandospezifische Daten. Für individuelle Kommandos welche lediglich einen Teilnehmer adressieren, wird dessen ID in den ersten beiden Datenbytes übermittelt. Eine 16 Bit Cyclic Redundancy Check (CRC) Prüfsumme schließt das Paket ab. Kontrolldatenpakete werden von der Vermittlungsstelle im Downlink grundsätzlich in allen 14 spektralen Gruppen identisch ausgesandt. Damit ist gewährleistet, dass jeder Teilnehmer unabhängig von seiner aktuell gewählten Gruppe die Daten erhält. Ein Teilnehmer sendet nur nach Aufforderung durch ein entsprechendes Kommando in einem Kontrollpaket im Downlink eine Antwort im Uplink zurück. Dazu fragt das OLT die Zustände der einzelnen ONUs regelmäßig sequenziell ab.

Die Übermittlung der eigentlichen Nutzdaten erfolgt im letzten und mit 8192 OFDM Symbolen auch größten Bereich eines Frames. Im Gegensatz zu den Kontrolldaten sind hier die Datenträger einer spektralen Gruppe, welche für die Übertragung zwischen OLT und ONUs genutzt werden, frei konfigurierbar. Weiterhin kann auch das Modulationsverfahren frei gewählt werden und es besteht die Möglichkeit, eine Übertragung auf einen beliebigen Bereich innerhalb der 8192 Symbole einzugrenzen. So ist es möglich, mehrere ONUs in der selben spektralen Gruppe zu betreiben, indem diese unterschiedliche Datenträger und/oder OFDM Symbolbe-

#### 4. Systemarchitektur

---

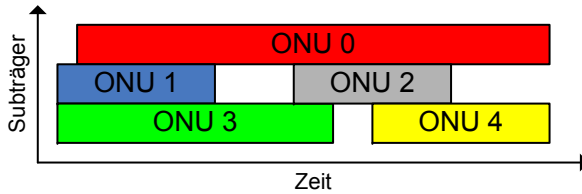


Abbildung 4.10.: Nutzdaten Konfigurationsoptionen in Zeit- und Frequenzbereich

reiche (ähnlich zu TDMA) nutzen. Entsprechend kann die Bandbreite für jeden Teilnehmer individuell im Downlink wie auch im Uplink gewählt werden. Abbildung 4.10 zeigt diese zweidimensionalen Konfigurationsoptionen grafisch und demonstriert gleichzeitig die Flexibilität dieses Ansatzes.

Die Einbettung der Nutzdaten erfolgt wie in Abbildung 4.11 dargestellt: Abhängig von der gewählten Konfiguration der aktiven Datenträger, deren Modulationsformat und des Symbolbereichs wird zwischen dem OLT und einer ONU ein Übertragungskanal gebildet. Die Anzahl der in diesem Kanal übermittelten Bits pro Frame kann je nach Konfiguration sehr unterschiedlich sein. Aus diesem Grund werden die Nutzdaten nicht wie bei den Kontrolldaten in Form eines Pakets innerhalb eines Frames übermittelt. Stattdessen werden die übermittelten Daten einzelner Frames nahtlos aneinander angehängt und bilden so einen *virtuellen Kanal*. Die Absicherung dieses Kanals erfolgt, ähnlich wie bei den Kontrolldaten, ebenfalls mit einem Reed-Solomon-Code. Anstatt einer Coderate von  $\frac{1}{2}$  kommt jedoch eine Rate von  $\frac{223}{255}$  zum Einsatz. Die Übermittlung der eigentlichen Nutzdatenpakete (sog. *virtuelle Kanalpakete*) erfolgt schließlich innerhalb dieses abgesicherten virtuellen Kanals.

Ein solches Paket beginnt stets mit einer konstanten Folge aus den vier Bytes 0x45, 0x3D, 0xCD, 0x28. Dieser sog. *Magic Code* ermöglicht das einfache Erkennen eines Paketanfangs, da es sehr unwahrscheinlich ist, dass diese Bytefolge genau so in den Nutzdaten auftritt. Sollte dieser Fall doch einmal auftreten, so wird ein falsches Paket dekodiert und übergeordnete Schichten müssen diesen Fall abfangen. Im Anschluss an diesen Code

## 4.2. Spezifikation der Systemparameter und Kommunikationsprotokolle

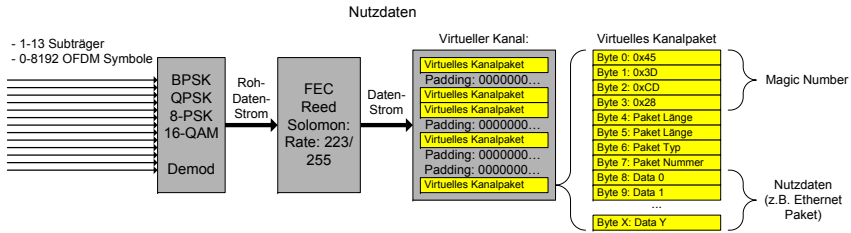


Abbildung 4.11.: Aufbau des Nutzdatenbereichs

wird die Länge des Pakets sowie dessen Typ und eine laufende Nummer übermittelt. Der darauf folgende Datenbereich beinhaltet dann die eigentlichen Nutzdaten. Das können beispielsweise Ethernet-, IP- oder Common Public Radio Interface (CPRI)-Pakete sein. Die Unterscheidung der unterschiedlichen Pakettypen erfolgt über den zuvor übermittelten Typ und die laufende Nummer ermöglicht die Entdeckung von Paketen, die aufgrund eines Übertragungsfehlers verloren gegangen sind. Diese *virtuellen Kanalpakete* werden in den virtuellen Kanal eingebettet. Sollte die Kanalbandbreite größer sein als für die zu übermittelnden Pakete benötigt, wird der Kanal mit einem festen Pattern von 0x00 aufgefüllt. So wird die Möglichkeit ausgeschlossen, dass ein Empfänger fälschlicherweise den Beginn eines neuen Pakets entdeckt während der Kanal unbesetzt ist.

Damit ist die Spezifikation der Systemparameter des OFDM Netzwerks abgeschlossen. Zusammenfassend sind alle wichtigen Parameter noch einmal in Tabelle 4.3 aufgelistet.

Auch der Overhead der OFDMA Prozessierung und des Kommunikationsprotokolls kann nun ermittelt werden und ist in den Tabellen 4.4 und 4.5 zusammengefasst.

Kombiniert man die Ergebnisse beider Tabellen, erhält man einen Gesamt-Overhead des Kommunikationssystems bzgl. der verfügbaren Kanalbandbreite von 51.2%. Auch die je nach Modulationsformat maximal erzielbaren Datenraten lassen sich nun berechnen (Tabellen 4.6, 4.7)

#### 4. Systemarchitektur

Parameter	Wert
OLT Samplerate DAC & ADC	25 GSa/s
OLT FFT Größe	256
OLT Gesamtdatendurchsatz (pro Richtung)	50 GBit/s
OLT Gesamtbandbreite	25 GHz
ONU Samplerate DAC & ADC	3.125 GSa/s
ONU FFT Größe	32
Cyclic prefix	25%
Länge eines OFDM Symbols (OLT/ONU)	$\frac{320}{40}$ Samples
Dauer eines OFDM Symbols	12.8 ns
Anzahl spektraler Gruppen	14
Datenträger pro spektraler Gruppe	13
Pilottöne pro spektraler Gruppe	2
Gesamtträger pro spektraler Gruppe	16
Anzahl Kontrollsymbole pro Frame	32
Anzahl Datensymbole pro Frame	8192
Gesamtsymbole eines Frames	8250
Dauer eines Frames	105.6 us

Tabelle 4.3.: Vollständige OFDM Systemparameter

Systemkomponente	Overhead
Zwei Pilottöne pro spektraler Gruppe	$\frac{28}{256} = 10.9375\%$
Ein DC Träger pro spektraler Gruppe	$\frac{14}{256} = 5.4688\%$
Ein DC Träger am OLT	$\frac{1}{256} = 0.39\%$
Zwei Unbenutzte spektrale Gruppen	$\frac{32}{256} = 12.5\%$
Cyclic prefix	20%
Gesamt-Overhead der physikalischen Schicht	43.4375%

Tabelle 4.4.: Overhead der OFDMA Prozessierung



## 4.2. Spezifikation der Systemparameter und Kommunikationsprotokolle

---

Systemkomponente	Overhead
10 Synchronisationssymbole	$\frac{10}{8250} = 0.1212\%$
16 Phasenreferenzsymbole	$\frac{16}{8250} = 0.1939\%$
32 Kontrolldatensymbole	$\frac{32}{8250} = 0.3879\%$
FEC & virtuelles Kanalpaket	13.1282%
Gesamt-Protokolloverhead	13.7389%

Tabelle 4.5.: Overhead des Kommunikationsprotokolls

Modulationsformat	Datenrate
BPSK	12.2653 GBit/s
QPSK	24.5305 GBit/s
8-PSK	36.7958 GBit/s
16-QAM	49.061 GBit/s

Tabelle 4.6.: Nutzdatenraten des Gesamtnetzwerks

Modulationsformat	Datenrate
BPSK	0.8761 GBit/s
QPSK	1.7522 GBit/s
8-PSK	2.6283 GBit/s
16-QAM	3.5044 GBit/s

Tabelle 4.7.: Nutzdatenraten pro spektraler Gruppe

### 4.3. Hardware Architektur

#### 4.3.1. Downlink

Nachdem in den beiden letzten Abschnitten die Spezifikation der Anforderungen an die Komponenten und der Systemparameter erfolgte, sowie eine Übersicht über die grundsätzliche Funktionsweise des Systems gegeben wurde, soll nun der Aufbau der digitalen Verarbeitungskette zur Realisierung des vorgestellten Netzwerks genauer betrachtet werden. Dazu wird zunächst die Realisierung der physikalischen Schicht des Downlinks betrachtet, anschließend folgt der Uplink und schließlich die Anbindung an die Logik der übergeordneten MAC-Layer.

Als Startpunkt des Downlinks im OLT kann der *Framecounter* betrachtet werden. Er generiert kontinuierlich die im vorigen Abschnitt vorgestellte Framestruktur, indem drei Signale erzeugt werden, welche den aktuellen Abschnitt im Frame, den Symbolindex im aktuellen Abschnitt und den Sampleindex im aktuellen Symbol bereitstellen. Diese *Zeitbasis* wird von vielen weiteren Komponenten im OLT genutzt. Abbildung 4.12 zeigt einen schematischen Aufbau der physikalischen Schicht des OLT Downlink Transmitters.

Basierend auf der Zeitbasis des Framecounters wird eine Vielzahl von Kontrollsignalen generiert. Diese steuern unter anderem einen Multiplexer, welcher basierend auf dem aktuellen Frameabschnitt entweder passende Konstanten für die Erzeugung von Sync-Sequenz und Phasenreferenz erzeugt, oder Kontroll- und Nutzdaten über ein Interface von der MAC Layer bezieht und an einen Modulator weiterleitet. Der Modulator generiert aus den eintreffenden Datenbits entsprechende komplexe Konstellationssymbole basierend auf dem Konstellationsdiagramm der gewählten Modulationsart. Während für Kontrolldaten immer eine BPSK Modulation verwendet wird, besteht für Nutzdaten eine Auswahl zwischen BPSK, Quadrature-Phase-Shift-Keying (QPSK), 8-PSK und 16-QAM. Abbildung 4.13 zeigt diese vier unterstützten Modulationsformate. Welches Format zum Einsatz kommt, wird dem Modulator parallel zu den eigentlichen Nutzdaten vom MAC Layer mitgeteilt, so dass es jederzeit innerhalb des Nutzdatenbereichs eines Frames möglich ist, das Modulationsformat zu ändern. Eine solche Situation kann beispielsweise

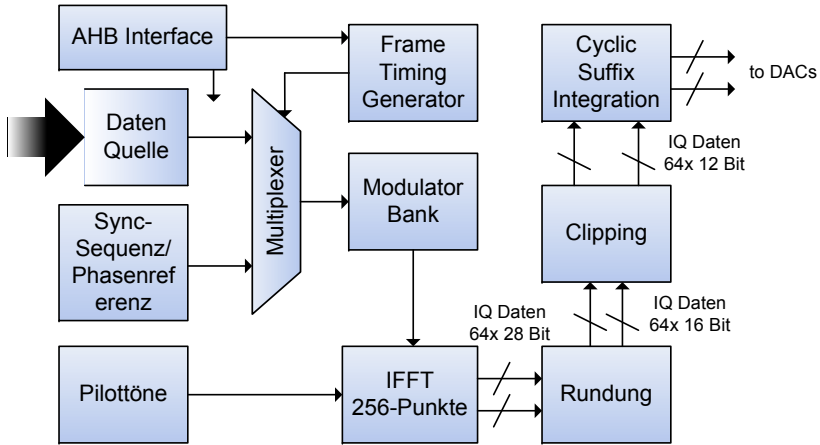


Abbildung 4.12.: OLT Transmitter Architektur

dann auftreten, wenn mehrere Abschnitte innerhalb des Nutzdatenbereichs für unterschiedliche ONUs vorgesehen sind und sich die gewählte Modulationsart zwischen den ONUs aufgrund unterschiedlicher Signal-Störabstände unterscheidet.

Dem Modulator schließt sich die eine 256 Punkte IFFT an, welche das bisher im Frequenzbereich erzeugte Downlink Signal in den Zeitbereich umsetzt. Neben den Symbolen der Datenträger, welche vom Modulator erzeugt wurden, werden dem Eingangsvektor der IFFT noch die

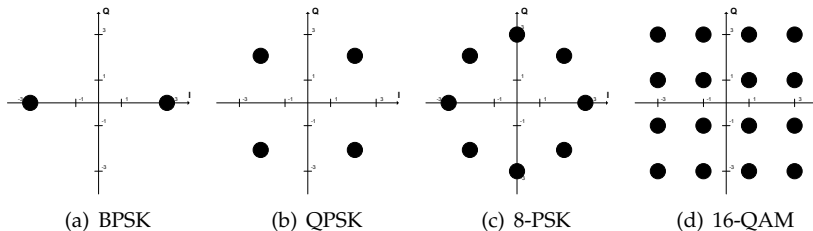


Abbildung 4.13.: Unterstützte Modulationsformate

konstanten Pilottöne für die einzelnen spektralen Gruppen hinzugefügt. Außerdem werden die dauerhaft unmodulierten Subträger mit einer konstanten Null belegt. Die Berechnung der IFFT ist recht aufwändig und somit ressourcenintensiv. Implementierungsdetails sind in Kapitel 5.2 zu finden.

Die Elemente des Ausgangsvektors der IFFT besitzen eine Auflösung von 14 Bit. Da die verwendeten DA Wandler jedoch lediglich eine Auflösung von 6 Bit erlauben, wird das Zeitbereichssignal in den nächsten beiden Schritten zunächst auf eine konfigurierbare Anzahl von Nachkommastellen gerundet und anschließend mit Hilfe einer Clippingeinheit seiner Signalspitzen beraubt, indem die höchstwertigen beiden Bits entfernt werden. Das Entfernen der Signalspitzen reduziert das hohe PAPR von OFDM Signalen deutlich und hat nur wenig Einfluss auf die Performanz der Übertragung, wie in [23] nachgewiesen wurde. Durch die Wahl des Verhältnisses zwischen Rundung und Clipping lässt sich die Qualität des Ausgangssignals des DA Wandlers optimieren. In einer letzten Stufe vor der Ausgabe wird dem Signal noch der Cyclic Prefix, bzw. in der konkreten Implementierung dieser Arbeit in Form eines Cyclic Suffix, hinzugefügt.

Das komplexe, digitale Basisbandsignal des OLT wird nun von zwei 25 GSa/s DA Wandlern in ein analoges Signal umgesetzt, welches anschließend wie zu Beginn dieses Kapitels dargestellt zur Modulation des Downlink-Lasers verwendet wird.

Nachdem das Downlink Signal die optische Netzwerkinfrastruktur passiert hat und das analoge Frontend einer ONU eine spektrale Gruppe herausgefiltert und ins komplexe Basisband herunter gemischt hat, wird das Signal wieder von zwei 3.125 GSa/s AD Wandlern digitalisiert. Abbildung 4.14 illustriert die Struktur des ONU Empfängers, der das analog gefilterte, komplexe Basisbandsignal der elektrischen Mischerstufen an den Eingängen der ADCs entgegennimmt. Anders als beim OLT ist hier die Abtastfrequenz der Wandler in einem gewissen Rahmen variabel, um wie in Abschnitt 4.2.4 beschrieben auf Frequenzoffsets reagieren zu können. Der Taktgenerator zur Versorgung der Wandler ist daher in Form einer DDS mit anschließender PLL ausgeführt wie in Abschnitt 6.4 noch erläutert wird. Das digitalisierte Basisbandsignal passiert im ONU Empfänger zunächst eine Frequenzoffsetkorrektur. Diese besteht

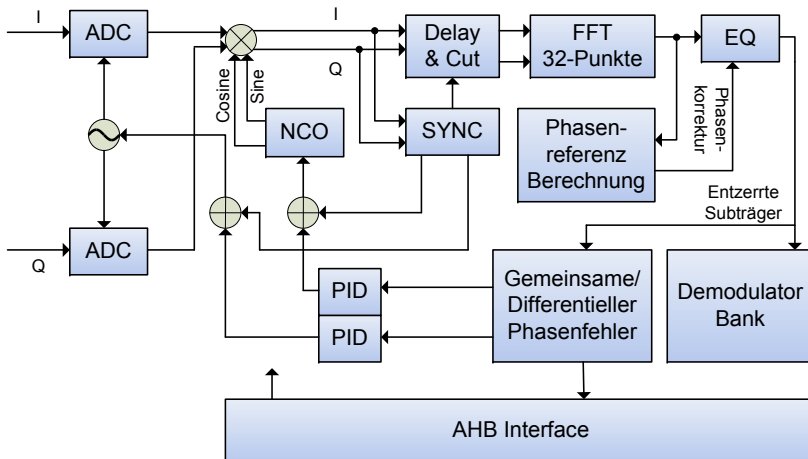


Abbildung 4.14.: ONU Empfänger Architektur

aus einem NCO, dessen komplexes Ausgangssignal mit dem komplexen Basisbandsignal multipliziert wird. Auf diese Weise wird die Phasenlage aufeinander folgender Empfangssamples kontinuierlich rotiert, so dass auch leichte Frequenzfehler des Lokoszillators im analogen Frontend des ONU Empfängers kompensiert werden können.

Das nun von Frequenzoffsets befreite Empfangssignal wird weiter zu einem *Synchronisationsmodul* geleitet. Hier findet die Detektion der Sync-Sequenz und die Ermittlung der initialen Frequenzoffsets statt (vgl. Abschnitt 4.2.4). Abbildung 4.15 zeigt den inneren Aufbau dieses Moduls.

Das empfangene Signal wird in zwei parallelen Zweigen ausgewertet: Zum einen wird eine Autokorrelation mit einem Versatz von 200 Samples (fünf OFDM Symbolhöhen) durchgeführt und zum anderen wird die Signalenergie ermittelt. Beide Werte werden für eine Länge von 400 Samples (zehn OFDM Symbolhöhen) errechnet und anschließend verglichen. Dazu wird das komplexe Korrelationsergebnis mittels eines CORDIC Algorithmus in Polarkoordinaten transformiert, so dass die Korrelationsamplitude direkt mit der Signalenergie verglichen werden kann. Dieser Vergleich ist sehr wichtig, da die Korrelationsamplitude nicht nur

## 4. Systemarchitektur

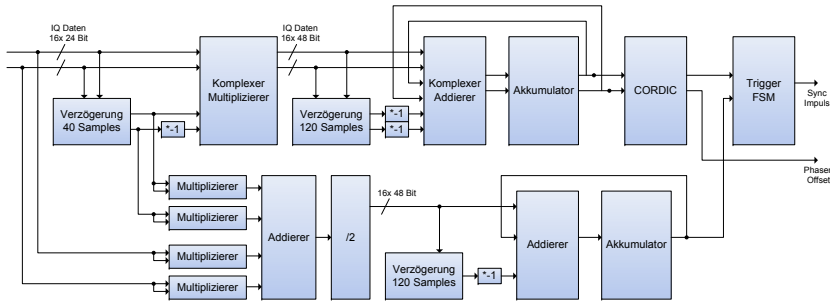


Abbildung 4.15.: ONU Synchronisationsmodul

von der Struktur des Empfangssignals abhängt, sondern auch von dessen Empfangsleistung. Relevant für die Detektion der Sync-Sequenz ist jedoch lediglich die Signalstruktur. Eine entsprechend designte Finite State Machine (FSM) prüft die Signalstruktur auf eine erwartete Spitze mit korrekter Anstiegs- und Abfallzeit und emittiert bei Detektion einen Sync-Impuls. Gleichzeitig kann das Ergebnis der Phasenberechnung des CORDIC Algorithmus zum Synchronisationszeitpunkt direkt für die Schätzung des Frequenzoffsets des Lokaloszillators verwendet werden. Somit sind alle für die Empfangssignalsynchronisation und Frequenzkorrektur benötigten Informationen verfügbar.

Die Messergebnisse des Synchronisationsmoduls werden in einem weiteren Schritt auf Plausibilität geprüft und es werden entsprechende Frequenzkorrektursignale für die Oszillatoren generiert. Liegen die ermittelten Frequenzfehler unter einer konfigurierbaren Schwelle, so werden die aktuellen Korrekturwerte gelockt und die Verarbeitung des Empfangssignals im Frequenzbereich gestartet. Dazu erhält das *Delay&Cut Modul* vom Synchronisationsmodul die Startposition eines neuen Frames. Das Modul entnimmt nun aus dem kontinuierlich eintreffenden Strom von Abtastwerten im Abstand von 40 Samples (OFDM Symbollänge) Gruppen von je 32 Samples (im ONU verwendete FFT Größe). Aufgrund des am OLT hinzugefügten Cyclic Suffix von acht Samples muss der Entnahmezeitpunkt nicht aufs Sample genau stattfinden. Er sollte jedoch möglichst in der Mitte eines Symbols liegen, um Verschiebungen zu beiden Enden des Symbols bestmöglich zu tolerieren. Parallel dazu wird ein

Framecounter ähnlich dem im OLT Transmitter auf das Empfangssignal synchronisiert und liefert ab diesem Zeitpunkt eine genaue Zeitbasis für weitere Signalverarbeitungs-komponenten.

Anschließend an die Entnahme der Samplegruppen erfolgt die Berechnung einer 32 Punkte FFT, die das Empfangssignal zur weiteren Verarbeitung in den Frequenzbereich transformiert. Die verwendete FFT basiert auf der selben Implementierung wie auch die 256 Punkte FFT des OLT (siehe Kapitel 5.2).

Der erste Verarbeitungsschritt nach der FFT ist die Phasenreferenzberechnung. Hier wird die Phasenreferenzsequenz des empfangenen Frames ausgewertet und anhand der bekannten Ursprungssymbole können Korrekturfaktoren für die einzelnen Subträger ermittelt werden. Dazu erfolgt für jeden Subträger eine Division des Sollwertes durch den Empfangswert. Da der Sollwert bekannt ist und  $1+0j$  entspricht, vereinfacht sich die Division zu einer Inversion. Die ermittelten Korrekturwerte werden schließlich an ein *Equalizermodul* weitergeleitet, welches im Folgenden sämtliche Träger mit den komplexen Korrekturwerten multipliziert und somit Phase und Amplitude möglichst auf Werte korrigiert, die dem Zustand bei deren Aussendung entsprechen.

In einem letzten Schritt werden die beiden Pilotöne aus dem Spektrum extrahiert und der CPE und DPE wie in Abschnitt 4.2.4 dargestellt ermittelt. Die beiden Offsetfehler werden im Weiteren an zwei PID Controller weitergeleitet, die schließlich auf die Stellglieder zur Kompensation der Frequenzoffsets wirken.

Sämtliche Datenträger werden hingegen zu einem Demodulator geleitet, welcher die komplexen Symbole basierend auf einem vom MAC Layer gewählten Modulationsverfahren wieder zurück in die ursprünglichen Kontroll- und Nutzdaten verwandelt.

### 4.3.2. Uplink

Während der Downlink eigenständig synchronisiert wird und damit auch unabhängig vom Uplink arbeiten kann, benötigt dieser in der hier vorgestellten Realisierung zwingend einen parallel arbeitenden Downlink. Von dort bezieht er die Korrekturwerte für die Lokal- und Samplingos-

## 4. Systemarchitektur

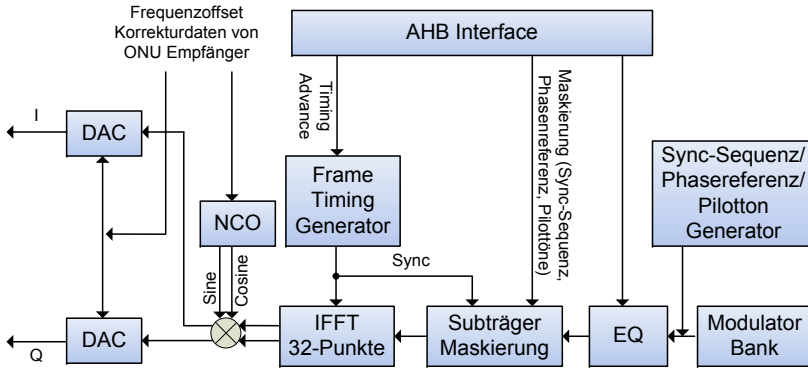


Abbildung 4.16.: ONU Transmitter Architektur

zillatoren sowie über Kommandos, welche vom Downlink transportiert werden, *Timing Advance* und *Pre-Equalization* Parameter. Abbildung 4.16 zeigt den schematischen Aufbau des Uplink Transmitters an einer ONU.

Wie auch beim OLT wird die *Zeitbasis* innerhalb des ONU Senders von einem Framecounter generiert. Dieser arbeitet jedoch nicht fix, sondern erlaubt eine beliebige zeitliche Verschiebung des Sendesignals mit der Granularität von einem Sample. So können sämtliche im Netzwerk vorhandenen ONUs so konfiguriert werden, dass sich ihre Signale am OLT zeitlich exakt überlagern. Die Kontroll- und Nutzdaten, die von einer ONU ans OLT gesendet werden müssen, werden in einem Modulator in komplexe Symbole transferiert. Auch hier stammt die Wahl des Modulationsformats direkt von der übergeordneten MAC Layer. Ebenfalls werden die beiden Pilotöne sowie in den ersten beiden Frameabschnitten die Symbole für die Sync-Sequenz und die Phasereferenz dem Subträgervektor in Form von konstanten Symbolen hinzugefügt. Anders als beim OLT erfolgt nun jedoch nicht sofort die Transformation in den Zeitbereich, sondern es folgt zunächst eine *Pre-Equalization* und eine *Subträgermaskierung*. Die *Pre-Equalization* korrigiert Phase und Amplitude der ausgesendeten Träger bereits vorab im Sender so, dass sie beim OLT in korrekter Phaselage und mit der richtigen Amplitude eintreffen. Dieses Vorgehen bietet sich an, da es zum einen den Verarbeitungsaufwand am OLT maßgeblich reduziert und zum anderen den deutlich höheren Dynamikbereich der



DA Wandler einer ONU gegenüber den AD Wandlern des OLT effektiv ausnutzt. Die *Subträgermaskierung* sorgt dafür, dass ein Träger einer ONU nur dann moduliert wird, wenn dies auch vom OLT mit Hilfe eines Kommandos über den Downlink angefordert wurde. So wird beispielsweise verhindert, dass sich Sync-Sequenzen oder Phasenreferenzen mehrerer ONUs gegenseitig überlagern und damit einen klaren Empfang am OLT unmöglich machen.

Anschließend folgt schließlich eine 32 Punkte IFFT, die das Signal vom Frequenzbereich in den Zeitbereich umsetzt. Die folgenden Module zum Runden und Clippen des Ausgangssignals arbeiten analog zu ihren Gegenstücken im OLT Sender. Schließlich wird auch hier noch ein Cyclic-Suffix eingefügt und das Signal ist im Prinzip sendebereit. Um das OLT jedoch auch von der Aufgabe der Frequenzoffsetkorrektur der Sample- und Lokaloszillatoren zu befreien, wird dem Ausgangssignal in einem letzten Schritt mit Hilfe eines NCO und eines komplexen Multiplikators noch ein Frequenzoffset hinzugefügt, der dem inversen ermittelten Frequenzfehler des Downlink Empfängers entspricht. Da die IQ-Mischer von Downlink und Uplink sowohl am OLT als auch am ONU einen gemeinsamen Lokaloszillator verwenden und die Signalausbreitungsbedingungen aufgrund der Verwendung von Glasfaserkabeln sehr stabil sind, lässt sich so die Frequenzoffsetkorrektur im Uplink ressourcenschonend ohne direkte Messung am OLT realisieren.

Wie in Abbildung 4.17 gut zu erkennen ist, kann der OLT Empfänger recht einfach aufgebaut werden. Es werden keine Module für eine Frequenz-, Phasen- oder Zeitkorrektur benötigt, da diese Aufgaben bereits vom ONU Sender erledigt werden. Was es jedoch noch gibt ist ein (recht einfach aufgebautes) *Sync-Modul*, und einen *Phasenreferenzrekorder*. Das Sync-Modul führt eine komplexe Kreuzkorrelation mit dem Empfangssignal und einer Referenzsinusschwingung durch und detektiert so die von den ONUs ausgesandten Sync-Pulse. Da die Kreuzkorrelation nur einmal pro empfangenen OFDM Symbol durchgeführt wird, hält sich der Rechenaufwand in Grenzen. Um die genaue Lage eines empfangenen Sync-Pulses bezüglich des OLT Framecounters zu bestimmen, kann die Phase des Korrelationsergebnisses ausgewertet werden. Das Ergebnis der Berechnung wird vom OLT Prozessorsystem über ein Businterface ausgelesen und entsprechende Kontrollsignale zur Korrektur des Startzeitpunkts des Frames an die betroffene ONU gesandt. Der *Phasenreferenzrekorder* hat die

## 4. Systemarchitektur

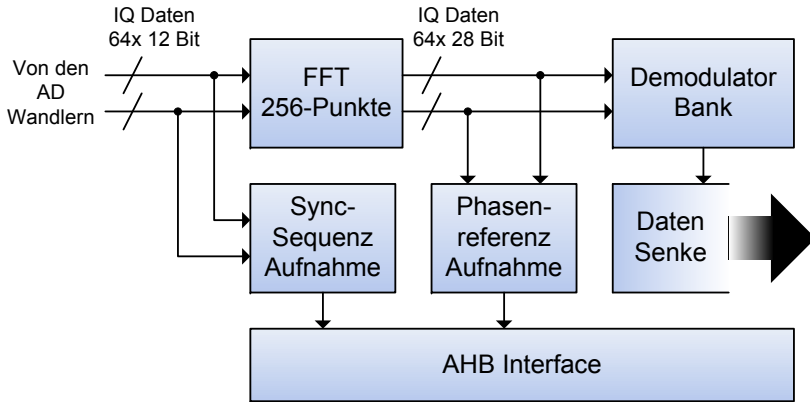


Abbildung 4.17.: OLT Empfänger Architektur

Aufgabe, die komplette empfangene Phasenreferenz zu speichern und über das Businterface dem Prozessorsystem zur Verfügung zu stellen. Die Phasenlagen der einzelnen empfangenen Subträger werden dann in Software ausgewertet und passende Korrekturwerte berechnet. Mit Hilfe einer Kontrollnachricht werden diese schließlich im Downlink zur betroffenen ONU gesendet, so dass die Uplink *Pre-Equalizer* entsprechend konfiguriert werden können.

Der Rest des Uplinkempfängers des OLT besteht im Wesentlichen noch aus einer 256 Punkte FFT, der sich direkt ein Demodulator zur Umwandlung der komplexen Symbole zurück in die ursprünglichen Kontroll- und Nutzdaten anschließt.

### 4.3.3. Nutzdatentransfer und dynamische Bandbreitenallokation

In Abschnitt 4.2.5 wurde bereits erläutert, wie Nutzdaten mit Hilfe eines *virtuellen Kanals* in die Datenbereiche der Frames eingebettet werden. In diesem Abschnitt geht es nun darum, wie diese Einbettung für den in dieser Arbeit konkret demonstrierten Fall von 10G Ethernet funktioniert.

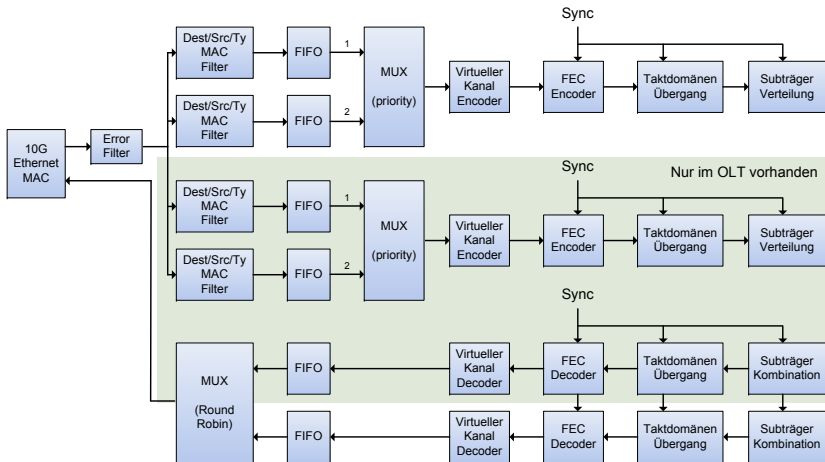


Abbildung 4.18.: MAC Layer

Abbildung 4.18 zeigt die Architektur der Ethernet Paketdatenverarbeitung, auch MAC Layer genannt. Die grundsätzliche Funktion ist dabei für OLT und ONU identisch, wobei im OLT jedoch für jede am Netz angeschlossene ONU ein eigener Verarbeitungszweig integriert wird. Für die Anwendung in einem produktiven Zugangsnetzwerk ist dieser Ansatz eigentlich ungeeignet. Einerseits müssen die einzelnen Verarbeitungszweige bei dieser Art der Implementierung auf den maximal möglichen Durchsatz einer ONU ausgelegt sein. Andererseits müssen auch genügend Zweige für die Maximalanzahl parallel genutzter ONUs am OLT zur Verfügung stehen. In der Praxis schließen sich jedoch beide Fälle gegenseitig aus, denn wenn im Netzwerk sehr viele ONUs angemeldet sind, können nicht alle gleichzeitig mit ihrer maximalen Bandbreite operieren. Aus diesem Grund wäre für das OLT eine dynamisch adaptive Struktur deutlich besser geeignet, welche die verfügbaren Ressourcen einerseits für ONUs mit hohem Durchsatz bündelt aber andererseits auch in der Lage ist, im Fall von einer hohen Anzahl an angemeldeten ONUs die verfügbaren Ressourcen gleichmäßig auf viele parallel arbeitende Paketverarbeitungszweige zu verteilen.

Das in dieser Arbeit präsentierte Netzwerk dient jedoch lediglich zu Demonstrationszwecken und der Fokus liegt primär auf der Realisierung der physikalischen OFDM Schichten. Daher werden maximal zwei ONUs unterstützt, so dass für diesen Fall der gewählte Ansatz einer statischen Implementierung deutlich einfacher zu realisieren war. Abbildung 4.18 kann daher für die Beschreibung der MAC Layer im OLT und in den ONUs verwendet werden, wenn berücksichtigt wird, dass der grau hinterlegte Bereich ausschließlich im OLT vorhanden ist.

Die Begriffe Up- und Downlink sind in diesem Fall zur Beschreibung ungeeignet, da die vorgestellte Architektur als Teil beider Links zum Einsatz kommt. Stattdessen werden im Folgenden die Begriffe Empfangspfad und Sendepfad verwendet, die jeweils die Funktionsweise des präsentierten Pfades bezüglich der 10G Ethernet Schnittstelle beschreiben. Der Empfangspfad beginnt also am Ethernet Interface und leitet die Nutzdaten nach der Verarbeitung zu den Modulatoren des Up- bzw. Downlinks. Analog dazu nimmt der Sendepfad die Daten der Demulatoren entgegen und leitet sie zum Ethernet Interface.

### 4.3.3.1. Empfangspfad

Sowohl OLT als auch ONU besitzen zur Kommunikation mit den umgebenden Netzwerken eine optische 10G Ethernet Schnittstelle wie bereits in Abschnitt 4.1.3.1 erläutert wurde. Die physikalische Schicht dieser Schnittstelle wird weitestgehend mit Hilfe eines GTH Transceivers und etwas FPGA Logik direkt auf dem Chip realisiert. Sie bietet eine standardisierte 10 Gigabit Media Independent Interface (XGMII) Schnittstelle zur Anbindung an eine Ethernet MAC Schicht. Hierzu wurde ein 10G Ethernet MAC Controller von *OpenCores* [30] verwendet. Die Aufgabe dieses Controllers ist der Empfang und die Erzeugung elementarer Ethernetpakete inklusive Präambel und CRC. Aufgrund der hohen Datenrate verfügt der gewählte MAC Controller über je ein FIFO Interface für Empfangs- und Senderichtung mit einer Datenbusbreite von 64 Bit. Der Start bzw. das Ende eines Frames wird mit Hilfe von zwei Zusatzsignalen während des FIFO Zugriffs signalisiert.

Der 10G Ethernet MAC Controller ist als Block ganz links in Abbildung 4.18 dargestellt. Im Empfangspfad folgt als erstes ein sog. *Error Filter*

Modul, welches die Aufgabe hat, sämtliche vom Controller empfangenen Ethernetpakete auf Gültigkeit zu überprüfen. Dazu wird insbesondere sichergestellt, dass die empfangenen Pakete die im Ethernetstandard festgelegte Mindestlänge von 64 Bytes besitzen und eine Maximallänge von 1500 Nutzbytes nicht überschreiten. Weiterhin entfernt das Modul eventuelle Unterbrechungen von gültigen Datenwörtern beim Auslesen des FIFOs, wie sie beim Empfang der Daten vom GTH Transceiver aufgrund unterschiedlicher Taktdomänen entstehen können. Ziel ist es, dass die vom *Error Filter* Modul weitergeleiteten Pakete exakt einem vorgegebenen Format entsprechen und eventuelle Übertragungsfehler der 10G Ethernet Verbindung keinerlei undefinierten Zustände in den nachfolgenden Modulen hervorrufen können.

Das gefilterte Empfangssignal wird nun in jedem Verarbeitungszweig parallel auf zwei *MAC Filter* geleitet. Hier besteht die Möglichkeit, Pakete herauszufiltern, deren Ziel- oder Quelladresse, bzw. deren Pakettyp nicht einer Vorgabe entsprechen. Dazu werden die drei Parameter mit einem konfigurierbaren Filterwert verglichen und anschließend mit einer Bitmaske kombiniert. Ein Paket kann nur dann passieren, wenn alle drei Tests ein positives Ergebnis liefern. Mit Hilfe dieses Filtermoduls ist es möglich, gewisse Pakete zu priorisieren, z.B. IP-TV Daten oder Voice over IP (VoIP). Jeder Filter besitzt nachfolgend einen eigenen Paketpuffer (FIFO), welcher zur Zwischenspeicherung eintreffender Pakete genutzt wird. Die Tiefe der Puffer ist individuell konfigurierbar. Sollte der nachfolgende Up- bzw. Downlink nicht in der Lage sein, alle eintreffenden Ethernetpakete rechtzeitig entgegenzunehmen, werden Pakete verworfen falls der Puffer vollläuft.

Es ist die Aufgabe einer auf den Prozessorsystemen des OLT bzw. ONU ablaufenden Software kontinuierlich den Bandbreitenbedarf der virtuellen Kanäle zu erfassen und bei Bedarf die Kapazität des Kanals mit Hilfe einer Rekonfiguration der gewählten Subträgerallokation anzupassen. Dieser Vorgang wird auch als dynamische *Bandbreitenallokation* bezeichnet. Um der Software die Möglichkeit zu geben den aktuellen Datenfluss zu erfassen, befinden sich hinter den *MAC Filtern* Paket- und Datenzähler, welche zur besseren Darstellbarkeit nicht in Abbildung 4.18 eingezeichnet sind. Weiterhin kann der Füllstand der Paketpuffer ausgelesen werden.

Die Pakete aus den Puffern werden nun mit Hilfe eines Multiplexers in einen gemeinsamen Paketstrom kombiniert. Dabei arbeitet der Multiplexer im Prioritätsmodus. D.h. die Paketes eines Puffers haben immer Vorrang vor den Paketen des anderen Puffers. Auf diese Weise wird die Priorität der beiden Filterpfade definiert. Bei Bedarf ist es auch leicht möglich mehr als zwei Prioritätsebenen zu verwenden, indem weitere *MAC Filter* und Paketpuffer vor einem Multiplexer eingefügt werden. Anschließend erfolgt im *Virtuellen Kanalencoder* die Einbettung der Ethernetpakete in die virtuellen Kanalpakete. Der *Virtuellen Kanalencoder* liefert daher einen kontinuierlichen Datenstrom mit eingebetteten Paketen, der bei Bedarf wie in Abschnitt 4.2.5 beschrieben mit Nullen aufgefüllt wird.

Anschließend wird dieser Datenstrom mit Hilfe des *FEC Encoders* gesichert. Der verwendete FEC Algorithmus (Reed Solomon) stammt ebenfalls von *OpenCores* [31] und arbeitet aufgrund seiner konfigurierten Rate von  $\frac{223}{255}$  immer auf Datenblöcken von 223 Wörtern. Es ist daher notwendig, den FEC Encoder und Decoder eines virtuellen Kanals nach jeder Modifikation der Subträgerallokation neu zu synchronisieren. Dies geschieht genauso wie die Trägerallokation selbst mit Hilfe geeigneter Kontrollnachrichten.

Nach der Sicherung des Datenstroms mit Hilfe der FEC ist dieser im Prinzip bereit für die Aufteilung auf die einzelnen verwendeten Subträger. Die Aufteilung der Daten erfolgt eng verzahnt mit den eigentlichen Modulatoren, die sich, bedingt durch die Hardwarearchitektur, jedoch in der Taktdomäne der physikalischen Schicht der OFDM Signalverarbeitung befinden. Der Rest der zuvor beschriebenen Paketdatenverarbeitung erfolgt in der Taktdomäne des 10G Ethernet Interfaces. Aus diesem Grund wird der Datenstrom vor seiner Aufteilung auf die Subträger zunächst mit Hilfe eines *Taktdomänen Übergangs* in die Taktdomäne der OFDM Verarbeitung transferiert.

Die eigentliche Aufteilung des 64 Bit breiten, kontinuierlichen Datenstroms auf die einzelnen Subträger erfolgt schließlich im Modul *Subträger Verteilung*. Dieses Modul wird entsprechend der aktuell gewählten Subträgerallokation von der auf dem Prozessorsystem laufenden Software konfiguriert und liefert den OFDM Modulatoren die entsprechenden Datenströme sowie das gewählte Modulationsformat. Je nach gewähltem Format werden ein bis vier Bit pro Symbol von den Modulatoren

verarbeitet. Weiterhin muss das Modul beachten, dass eventuell nicht alle 13 Datensubträger einer spektralen Gruppe belegt sind, und ebenfalls möglicherweise weniger als 8192 OFDM Symbole im Nutzdatenbereich eines Frames ausgewählt wurden. Aufgrund dieser vielfältigen Auswahlmöglichkeiten ergeben sich viele mögliche Kombinationen für die Verteilung des eintreffenden 64 Bit breiten Datenvektors auf die einzelnen Modulatoren. Das Modul besitzt daher einen hohen Anteil an kombinatorischer Logik, welcher sich in der maximal erzielbaren Taktfrequenz niederschlägt. Diese Tatsache ist besonders gravierend, da es in der deutlich schnelleren OFDM Taktdomäne arbeitet. In kommerziellen OFDM Übertragungssystemen findet man vor der Aufteilung des Datenvektors auf die Subträger häufig noch eine Permutationseinheit. Diese hat die Aufgabe, die vom FEC Algorithmus gesicherten Bits über einen möglichst weiten zeitlichen und spektralen Bereich zu verteilen. So haben Bündelfehler weniger gravierende Auswirkungen auf die Fehlerkorrektur, da von jedem gesicherten Block nur wenige Bits betroffen sind. Auch diese Einheit wurden aufgrund des Demonstrationszwecks der Systemimplementierung weggelassen.

Damit ist die Paketdatenverarbeitung für den Empfangspfad abgeschlossen. Die Modulatoren der OFDM Sender nehmen die vom *Subträger Verteilungs* Modul bereitgestellten Daten entgegen und erzeugen das gewünschte OFDM Signal. Es soll hier noch angemerkt werden, dass in diesem Abschnitt lediglich die Verarbeitung der Nutzdaten behandelt wurde. Die Kontrolldatenpakete werden in einem eigenen Block erzeugt und ausgewertet. Aufgrund der im Verhältnis recht niedrigen Datenrate und der festen Paketstruktur kann dieser Block jedoch recht einfach und ressourcenschonend realisiert werden.

### 4.3.3.2. Sendepfad

Der Sendepfad kehrt im Wesentlichen die oben beschriebenen Schritte um und extrahiert die im virtuellen Kanal eingebetteten Ethernetpakete. Dazu liefert das *Subträger Kombinationen* Modul zunächst abhängig von der konfigurierten Subträgerallokation des virtuellen Kanals für jeden Träger das gewählte Modulationsformat an die Demulatoren im OFDM Empfänger. Eine festgelegte Anzahl von Takten später errei-

chen die demodulierten Datenströme der einzelnen Träger das *Subträger Kombinations* Modul und werden dort wieder zu einem 64 Bit breiten Datenvektor zusammengesetzt. Wie auch beim *Subträger Verteilungs* Modul müssen hierbei die gewählten Modulationsformate, die belegten Datenträger sowie die genutzten OFDM Symbole im Nutzdatenbereich des aktuellen Frames beachtet werden, so dass bei identischer Konfiguration der Subträgeraufteilung im Empfangspfad und der Subträgerkombination im Sendepfad identische Datenvektoren an der Schnittstelle zu den Taktdomänenübergängen resultieren.

Wie auch im Empfangspfad folgt vor der weiteren Paketverarbeitung zunächst der Übergang in die Taktdomäne des 10G Ethernet Interfaces. Anschließend wird mit Hilfe des FEC Decoders der empfangene Datenstrom auf Fehler überprüft und soweit möglich wiederhergestellt. Die Realisierung des FEC Decoders ist sehr ressourcenintensiv und aus diesem Grund wurde auch lediglich eine Variante mit *Hard-Decision* anstelle von *Soft-Decision* eingesetzt. Dabei wird ausschließlich der vom Demodulator ermittelte binäre Wert eines Symbols beachtet anstatt zusätzlich die Zuverlässigkeit, mit der die Demodulationsentscheidung getroffen wurde, ausgedrückt durch einen Fehlervektor (Error-Vector (EV)), mit in die Fehlerkorrektur einzubeziehen.

Der überprüfte und gegebenenfalls reparierte Datenstrom erreicht nun den *Virtuellen Kanaldecoder*. Hier wird der kontinuierlich eintreffende Datenstrom auf *Magic Codes* untersucht und bei Auftreten der Beginn eines virtuellen Kanalpakets erkannt. Bevor ein Ethernetpaket aus dem virtuellen Kanalpaket entnommen und weitergeleitet wird, erfolgt jedoch zunächst noch die Überprüfung der CRC des eingebetteten Ethernetpakets. Sollte die Korrektur des Datenstroms durch den FEC Decoder nicht erfolgreich gewesen sein, kann das an einer fehlerhaften Prüfsumme des Pakets abgelesen werden und es wird entsprechend verworfen.

Über Paketpuffer, die identisch zu denen im Empfangspfad arbeiten, gelangen die extrahierten Ethernetpakete im Falle des OLT zu einem Multiplexer. Dieser arbeitet nach dem *Round Robin* Verfahren und gewährt somit beiden Sendepfadzweigen identische Prioritäten. Anschließend wird das Ethernetpaket vom 10G Ethernet Controller entgegengenommen und schließlich über das 10G Interface versendet. Im Sendepfad einer ONU entfällt der Multiplexer, da hier nur ein Sendepfadzweig existiert.



## 4.4. Gesamtsystemübersicht

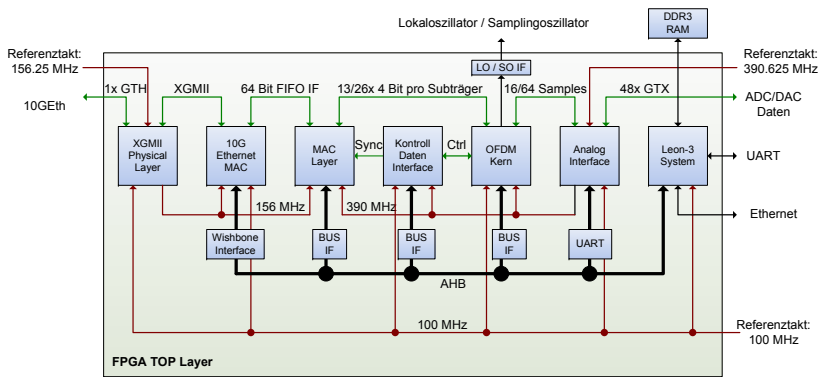


Abbildung 4.19.: FPGA Gesamtsystem

Auch die Paketpuffer müssen nicht allzu groß dimensioniert werden, denn die Bandbreite des 10G Ethernet Interfaces reicht im Normalfall leicht aus, um auch am OLT die Datenmengen von zwei voll ausgelasteten ONUs mit maximal je etwa 4 GBit/s zu abzuführen.

## 4.4. Gesamtsystemübersicht

Zum Schluss dieses Kapitels soll noch die Integration der zuvor vorgestellten Architekturkomponenten in ein Gesamtsystem vorgestellt werden. Abbildung 4.19 zeigt dazu eine Übersicht über das Top-Level FPGA Design des OLT bzw. einer ONU.

Die in Abschnitt 4.3.1 und 4.3.2 präsentierte Downlink- und Uplinkarchitektur wird durch den Block *OFDM Kern* repräsentiert. Die in Kapitel 4.3.3 vorgestellte Paketdatenverarbeitung verbirgt sich hinter dem Block *MAC Layer*. Im selben Kontext existiert noch der Block *Kontrolldaten Interface*, welcher für die Verarbeitung der Kontrolldaten verantwortlich ist. Die Blöcke *XGMII Physical Layer* und *Analog Interface* bilden die Schnittstelle des FPGA Designs nach außen. Aufgrund der sehr hohen transportierten Datenraten war ihre Realisierung besonders komplex und nahm einen großen Anteil des Aufwands in dieser Arbeit ein. Dabei realisiert der

#### 4. Systemarchitektur

---

Block *XGMII Physical Layer* die Anbindung der 10G Ethernet Schnittstelle an den FPGA, während der Block *Analog Interface* die Anbindung der 25 GSa/s bzw. 3.125GSa/s AD und DA Wandler beschreibt. Ihre Realisierung wird gesondert in Kapitel 6 beschrieben.

Ein weiterer Block, das *Leon-3 System*, kontrolliert die Datenverarbeitung in sämtlichen anderen Blöcken. Hierbei handelt es sich um ein System-on-Chip (SoC) Prozessorsystem von Gaisler Research [15], [13], [14]. Eine Sparc-V8 kompatible Central Processing Unit (CPU) mit 512 MByte DDR3-Systemspeicher ermöglicht den Betrieb eines *Embedded Linux* Betriebssystems. Über das Advanced High-performance Bus (AHB) Bussystem [3] ist die CPU mit sämtlichen weiteren Blöcken verbunden und hat somit die Möglichkeit, diese zu konfigurieren und zu überwachen. Die Kontrollsoftware ist auf mehrere Applikationen aufgeteilt und kontrolliert beispielsweise die Subträgerallokationen der Sender und Empfänger bzw. kümmert sich um die Verarbeitung der Kontrolldaten des Netzwerks. Die Verwendung eines Linux Betriebssystems erlaubt das einfache Speichern und Laden von Testdaten der Hardwaremodule in Dateien, die in einem weiteren Schritt auch sehr einfach und performant mit einem Personal Computer (PC) ausgetauscht werden können. Außerdem erlaubt die Unterstützung mehrerer Prozesse bzw. Threads durch das Betriebssystem eine einfache Aufteilung der unterschiedlichen Aufgaben der Kontrollsoftware in einfache Anwendungen.

## 5. Realisierung der Signalverarbeitung

In Kapitel 4 wurde die Hardwarearchitektur des in dieser Arbeit entwickelten Kommunikationssystems detailliert vorgestellt. In diesem Kapitel soll nun die eigentliche Realisierung der präsentierten Signalverarbeitungseinheiten genauer betrachtet werden. Viele der vorgestellten Komponenten, wie z.B. Addierer oder Multiplizierer, erscheinen auf den ersten Blick trivial. Bedenkt man jedoch, dass die DSP-Blöcke des OLT in jedem Takt 64 komplexwertige Samples bei einer Taktfrequenz von 390 MHz verarbeiten müssen, wird schnell klar, dass selbst die Realisierung einfachster digitaler Schaltkreise genau durchdacht und sorgfältig geplant werden muss. Hier findet sich auch die wesentliche Herausforderung dieser Arbeit, welche sich die Realisierung eines Systems an der Grenze des technisch Machbaren zur Aufgabe gemacht hat.

Aufgrund der Vielzahl verschiedener Hardwaremodule wird in diesem Kapitel lediglich eine Auswahl behandelt, die alle grundlegenden Prinzipien zur Optimierung der Performanz und Reduktion des Ressourcenbedarfs aufzeigt. Einen hohen Aufwand, sowohl bei der Konzeptionierung als auch bei der Realisierung, erforderte auch die Anbindung der DA und AD Wandler des OLT, da zum Aufbau des erforderlichen 300 GBit/s schnellen Datenlinks 48 MGTs gemeinsam koordiniert und synchronisiert werden müssen. Auch die Realisierung dieses Interfaces wird in diesem Kapitel detailliert behandelt.

### 5.1. Designkriterien für das FPGA System

Aufgrund des hohen Durchsatzes an Samples war bereits zu Beginn der Realisierung der DSP Module klar, dass das resultierende FPGA Design sehr ressourcenintensiv und timingkritisch wird. Um trotzdem bestmögliche Ergebnisse zu erzielen, wurden bereits zu Beginn einige Regeln definiert, welche ein möglichst performantes Design ermöglichen sollen:

- In ressourcenintensiven FPGA Designs wird der kritische Pfad primär durch Verzögerungen in der Verdrahtung (durch die sog. Switch-Boxen) und durch hohe Auffächerung (engl. Fan-Out) einzelner Netze bestimmt. Um trotzdem eine möglichst hohe Taktfrequenz zu erzielen, muss mit massivem Pipelining gearbeitet werden. Das gilt auch dann, wenn in einem Signalpfad nur wenig oder gar keine kombinatorische Logik zum Einsatz kommt, um so die Verdrahtung langer Netze bzw. von Netzen mit hohem Fan-Out zu optimieren. Hilfreich kann es hierbei sein, vor oder nach einem komplexen Modul einige Registerstufen in den Datenpfad einzufügen. Das erleichtert dem Retiming Algorithmus des Synthesetools den Aufbau einer effizienten Pipelining Struktur.
- Die Verwendung spezialisierter Hardwareeinheiten im FPGA (beispielsweise DSP Blöcke) sollte soweit möglich vermieden werden. Verglichen mit der feingranularen Verteilung der Configurable Logic Block (CLB) Blöcke sind diese nur an verhältnismäßig wenigen Positionen im FPGA verfügbar. Dementsprechend reduzieren sich die Freiheitsgrade bei der Platzierung und Verdrahtung der Logik deutlich, was wiederum zu erhöhtem Routingaufwand und somit Verzögerungen führt.
- Komplexere Operationen, wie beispielsweise die Addition vieler Eingangswerte, müssen mit Hilfe von Pipelining in elementare Operationen unterteilt werden. In einigen Fällen ist es durch eine geschickte Wahl der Hardwarearchitektur auch möglich, komplexe Operationen wie z.B. Multiplikationen oder Divisionen durch einfachere zu ersetzen oder sogar ganz einzusparen wie wir im nächsten Abschnitt 5.2 noch sehen werden.

- Selbstverständlich sollte für jedes in einem ressourcen- und timing-kritischen Bereich verwendete Signal stets die minimal mögliche Bitbreite verwendet werden. Das gilt auch für Zwischenergebnisse innerhalb komplexerer Operationen wie z.B. während der Berechnung einer FFT.
- Resetsignale besitzen häufig einen sehr hohen Fan-Out, da sie mit sehr vielen Registern verbunden sind. Gerade in datenflussorientierten Anwendungen (beispielsweise bei der Realisierung der hier vorgestellten OFDM Signalverarbeitung) benötigen viele Module gar keinen Reset. Solange das Modul in einer Pipelinestruktur eine Operation nach der Anderen auf die Eingangsdaten anwendet und keine Rückkopplungen oder FSMs zum Einsatz kommen, ergibt sich nach der Durchlaufzeit der Daten automatisch ein definierter Zustand am Ausgang.
- Während der Konzeptionierung des Datenflussgraphen für die DSP Module sollte ein möglichst geradliniger Datenfluss gewählt werden, der möglichst keine Rückkopplungen oder Multiplexer enthält. Ist das nicht möglich, sollte versucht werden, diese Elemente auf einen lokalen Bereich zu beschränken. So werden lange Netze welche zu hohen Latenzen führen vermieden oder zumindest stark reduziert.
- Generierte Module sollten möglichst universell gehalten und flexibel mit Hilfe von *Generics* parametrisiert werden können. Auch wenn so im ersten Moment die Implementierung komplizierter wird, kann auf diese Weise die Wiederverwertbarkeit einzelner Module deutlich erhöht werden, was viel redundanten Code erspart, den es weder zu erstellen noch zu verifizieren oder warten gilt. Auf diese Weise konnten in dieser Arbeit viele DSP Blöcke wie z.B. ein NCO in der Vermittlungsstelle wie auch bei den Teilnehmern identisch eingesetzt werden, obwohl sich sowohl die Anzahl der parallel zu bearbeitenden Samples als auch deren Auflösung unterscheiden.
- Ebenfalls sollte der Einsatz von fertigen IP Cores ohne Zugriff auf den Quellcode vermieden werden. Ansonsten kann es passieren, dass z.B. bei einem Wechsel der zugrundeliegenden FPGA Technologie oder des Synthesetools der IP Core nicht länger unterstützt wird. Natürlich gibt es auch immer Cores wie z.B. eine komple-

## 5. Realisierung der Signalverarbeitung

---

xe CPU, deren Eigenentwicklung zu aufwändig wäre, jedoch lassen sich viele einfachere Funktionseinheiten wie beispielsweise ein CORDIC Algorithmus auch einfach selbst erstellen und stehen so frei parametrierbar auch für nachfolgende Projekte zur Verfügung.

- Zu guter Letzt sollte stets die Ausgabe des Synthesetools im Auge behalten und überprüft werden, ob der ermittelte Ressourcenbedarf und das geschätzte Timing zumindest grob den Erwartungen entsprechen. Hier können auch Änderungen am Quellcode zur Optimierung des Moduldesigns leicht überprüft werden. Auch fehlerhaftes Verhalten im Synthesetool lässt sich so leicht aufdecken. Während der Realisierung dieser Arbeit wurde beispielsweise festgestellt, dass ein Addierbaum in Pipelinestruktur in manchen Fällen vom Xilinx eigenen Synthesetool *XST* nicht optimal umgesetzt wird. Nach einem Wechsel zu den Tools von Synopsys trat dieses Problem nicht mehr auf und es wurden deutlich weniger Ressourcen benötigt.

Ein weiterer wichtiger Punkt, der vor Beginn der Modulrealisierung festgelegt werden sollte ist die Darstellung von Informationen im FPGA Design. Die Verwendung geeigneter Typen vereinfacht nicht nur die Verknüpfung der Signale mit elementaren Operationen, sondern sie erhöht auch die Übersichtlichkeit, da sofort klar ist, zu welchem Zweck ein Signal eines bestimmten Typs dient. Tabelle 5.1 führt die wesentlichen in dieser Arbeit verwendeten Signaltypen und deren Verwendung auf.

Typ	Einsatzzweck
<code>std_logic</code>	Binaere Kontrollsignale
<code>std_logic_vector</code>	Datenbusse, unstrukturierte Daten
<code>signed &amp; unsigned integer</code>	Abtastwerte, Berechnungsergebnisse Zähler, Indizierung von Arrays

Tabelle 5.1.: Verwendete Signaltypen

Wichtig zur Reduktion des Ressourcenbedarfs ist immer auch die Wahl einer geeigneten Bitbreite (bzw. eines geeigneten Bereichs bei *integer* Typen). Für die Darstellung von Abtastwerten wäre die Bibliothek *IEEE.Fixed\_Pkg* sogar noch besser geeignet. Sie wurde jedoch nicht eingesetzt, da ihr Sup-

port durch die Synthesestools zum Zeitpunkt des Beginns der Arbeit noch unsicher war. Stattdessen kommen die Typen *signed* und *unsigned* aus der Bibliothek *ieee.numeric\_std* zum Einsatz. Sie erlauben zwar nur eine Darstellung von Ganzzahlen, so dass die Skalierung bei Festkommawerten manuell korrigiert werden muss, allerdings ist so zumindest eine einfache Kombination von vorzeichenbehafteten und vorzeichenlosen Werten möglich.

## 5.2. Fast Fourier Transformation

Mit zu den wichtigsten Signalverarbeitungsaufgaben im Kommunikationssystem gehören die Berechnung der FFT und der IFFT. Glücklicherweise können beide Berechnungen mit demselben Algorithmus durchgeführt werden. Zur Berechnung der IFFT mit Hilfe eines FFT Algorithmus müssen lediglich die Signale der Real- und Imaginärteile am Eingang und Ausgang der Berechnung getauscht werden [11].

An die Algorithmen zur Berechnung der 256-Punkt bzw. 32-Punkt FFT in dieser Arbeit werden besonders hohe Anforderungen bzgl. des Durchsatzes gestellt. Sie müssen 64 bzw. 16 Samples in jedem Takt parallel verarbeiten. Ein Vorteil des FFT Algorithmus ist, dass es sich um einen in sich abgeschlossenen Algorithmus handelt. D.h. die Verarbeitung erfolgt nicht wie beispielsweise bei einer Autokorrelation auf einem kontinuierlichen Datenstrom, sondern auf einer exakt festgelegten Menge an Abtastwerten (256 bzw. 32). Damit spielen bei der Berechnung einer FFT vorhergehende oder nachfolgende Abtastwerte keine Rolle. Um den geforderten hohen Durchsatz zu erreichen, ergeben sich zwei grundsätzliche Realisierungsmöglichkeiten:

Zum einen können 64 FFT Algorithmen, die jeweils einen Abtastwert pro Takt verarbeiten, parallel betrieben werden. Dazu müssen die Abtastwerte vor der Verarbeitung zunächst korrekt auf die einzelnen Algorithmen aufgeteilt und nach der Berechnung wieder zusammengefasst werden. Der Vorteil dieser Realisierung besteht in der hohen Verfügbarkeit effizienter, bereits vorhandener Algorithmen für die sequenzielle Verarbeitung von Abtastwerten. Die zweite Möglichkeit ist die Realisierung eines einzelnen Algorithmus, der in der Lage ist, direkt 64 Abtastwerte pro Takt zu

## 5. Realisierung der Signalverarbeitung

---

verarbeiten. Zu Beginn der Arbeit war kein solcher Algorithmus bekannt und er müsste daher neu erstellt werden.

Um eine Entscheidung zu treffen, welcher Ansatz besser für die Realisierung in dieser Arbeit geeignet ist, lohnt sich ein Blick auf die Struktur des FFT Algorithmus. Wie wir bereits in Kapitel 2.2.2 gesehen haben, lässt sich die elementare Berechnungseinheit einer FFT, der sog. *Butterfly* Algorithmus, auf unterschiedliche Arten realisieren, die sich vor allem in der Anzahl der parallel verarbeiteten komplexen Eingangswerte unterscheiden. Im Fall des *Radix-2 Butterfly* werden jeweils zwei komplexe Eingangswerte additiv bzw. subtraktiv verknüpft und bilden so den Ausgangsvektor. Im Fall des *Radix-4 Butterfly* werden stattdessen gleich vier komplexe Eingangswerte zu vier komplexen Ausgangswerten verknüpft. Neben Addition und Inversion (die einer Rotation des komplexen Eingangswertes um 180 Grad entspricht) kommen zusätzlich noch Rotationen um 90 bzw. 270 Grad hinzu. Die Rotationen werden durch die Multiplikation des Eingangswertes mit der imaginären Einheit  $+j$  bzw.  $-j$  realisiert. Es ist leicht ersichtlich, dass auch die um 90 bzw. 270 Grad rotierten Eingangswerte unter Zuhilfenahme einer Vertauschung des Real- und Imaginärteils des Eingangssignals durch einfache Additionen bzw. Subtraktionen mit den restlichen komplexen Eingangswerten an den Ausgabepunkten verknüpft werden können:

$$(a + bj) * j = -b + aj$$

Speziell wenn die in Abbildung 2.8 dargestellte Struktur 1:1 in digitale Logik umgesetzt wird, ist eine *Radix-4* Implementierung vorteilhaft, da eine Vertauschung von Real- und Imaginärteil ohne zusätzlichen Ressourcenbedarf realisierbar ist. Es lassen sich auch noch höherwertige Algorithmen definieren, wie z.B. einen *Radix-8* oder einen *Radix-16 Butterfly*, diese haben jedoch den Nachteil, dass Rotationen um 45 bzw. 22,5 Grad benötigt werden, welche sich nicht mehr durch einfache Vertauschungsoperationen realisieren lassen. Aus diesem Grund bildet der *Radix-4 Butterfly* Algorithmus die effizienteste Wahl für eine direkte FPGA Implementierung und wurde entsprechend in dieser Arbeit eingesetzt.

Ein weiterer wichtiger Schritt bei der Berechnung einer FFT ist die Multiplikation mit den Drehfaktoren. Dabei handelt es sich um komplexe Konstanten mit einem Betrag von 1 und einer Phase, die je nach Position



des Faktors im Algorithmus variiert. An dieser Stelle wird ein bedeutender Vorteil eines Algorithmus erkennbar, der parallel auf 64 Abtastwerten arbeitet: In diesem Fall finden die Drehfaktormultiplikationen alle parallel statt und der Faktor ist an jeder Multiplikatorposition konstant. Bei einer sequenziellen Verarbeitung hingegen muss, je nach Verarbeitungsschritt, ein anderer Faktor gewählt werden. Da sich Konstantenmultiplikationen jedoch deutlich ressourcenschonender realisieren lassen als Multiplikationen mit Variablen und die Gesamtzahl der Multiplikatoren für beide Realisierungsansätze identisch ist (der sequenziell arbeitende Algorithmus muss 64-fach parallel implementiert werden), ergibt sich hier ein klarer Vorteil für den parallel arbeitenden Algorithmus. Beachtet man zusätzlich, dass auch das *Bit Reordering* am Ende der Berechnung in einer parallelen Hardwarestruktur deutlich einfacher durch kostenloses Umverdrahten realisierbar ist und das Aufteilen und Zusammenführen des Datenstroms auf die einzelnen sequenziellen Algorithmen ebenfalls entfällt, wird klar, warum eine parallele Realisierung der FFT in dieser Arbeit vorteilhaft ist und gewählt wurde.

### 5.2.1. Radix Butterfly

Die grundlegende Struktur des realisierten FFT Algorithmus orientiert sich direkt an Abbildung 2.9. Zunächst wurden die elementaren Operationen realisiert: Eine *Radix-4 Butterfly* Struktur sowie eine Konstantenmultiplikation. In beiden Fällen wurde eine Implementierung gewählt, bei der die Bitbreiten frei anpassbar sind.

Die Realisierung der *Radix-4* Struktur ist relativ trivial. Mathematisch müssen folgende acht Berechnung umgesetzt werden:

$$\begin{aligned}
 out_{re}(0) &= in_{re}(0) + in_{re}(1) + in_{re}(2) + in_{re}(3) \\
 out_{im}(0) &= in_{im}(0) + in_{im}(1) + in_{im}(2) + in_{im}(3) \\
 out_{re}(1) &= in_{re}(0) + in_{im}(1) - in_{re}(2) - in_{im}(3) \\
 out_{im}(1) &= in_{im}(0) - in_{re}(1) - in_{im}(2) + in_{re}(3) \\
 out_{re}(2) &= in_{re}(0) - in_{re}(1) + in_{re}(2) - in_{re}(3) \\
 out_{im}(2) &= in_{im}(0) - in_{im}(1) + in_{im}(2) - in_{im}(3) \\
 out_{re}(3) &= in_{re}(0) - in_{im}(1) - in_{re}(2) + in_{im}(3) \\
 out_{im}(3) &= in_{im}(0) + in_{re}(1) - in_{im}(2) - in_{re}(3)
 \end{aligned}$$

## 5. Realisierung der Signalverarbeitung

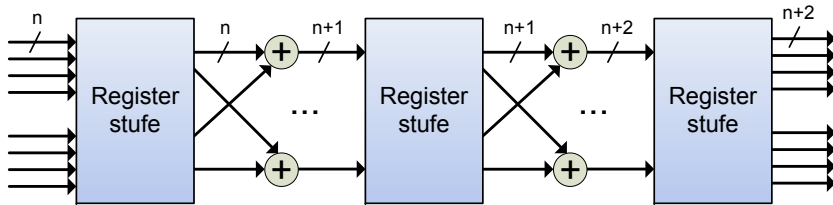


Abbildung 5.1.: Struktureller Aufbau eines Radix-4 Algorithmus

Es ist leicht erkennbar, dass pro Zeile vier Operanden addiert bzw. subtrahiert werden. Zur Maximierung der erzielbaren Taktfrequenz ist es daher sinnvoll, die Berechnung in zwei Stufen mit Additionen bzw. Subtraktionen von jeweils nur zwei Operanden durchzuführen. Zwischen den beiden Stufen sorgt eine Registerstufe für die Auftrennung der kombinatorischen Pfade und damit zur Erhöhung der möglichen Taktfrequenz. Weitere Registerstufen an den Ein- und Ausgängen haben dieselbe Funktion und sorgen so für eine Abgeschlossenheit der *Radix-4* Einheit. Auch die Bitbreite der involvierten Signale wird entsprechend der möglichen Berechnungsergebnisse nach jeder Addition oder Subtraktion um eins erhöht. Die Ausgangssignale besitzen entsprechend zwei Bit mehr als die Eingangssignale. Abbildung 5.1 illustriert den strukturellen Aufbau.

Mit Hilfe der Synthesetools *XST* von Xilinx bzw. *Synplify* von Synopsys wurde evaluiert, dass die gewählte Struktur die gewünschte hohe Taktrate bestmöglich erreicht. Tabelle 5.2 fasst die ermittelten Ergebnisse zusammen.

Parameter	XST(14.4)	Synplify (J-2015.03-SP1)
Anzahl Register	500	558
Anzahl LUTs	392	416
Maximale Taktfrequenz	758.438 MHz	862.7 MHz

Tabelle 5.2.: Synthesergebnisse der Radix-4 Butterfly Struktur  
(15 Bit breite Eingangsvektoren)

### 5.2.2. Drehfaktormultiplikationen

Zur Realisierung der Konstantenmultiplikation ist es hilfreich, zunächst einmal die Multiplikation wie in folgender Formel dargestellt zu betrachten:

$$\begin{aligned} \underline{z} &= \underline{x} * \underline{y} = (x_{re} + jx_{im}) * (y_{re} + jy_{im}) \\ z_{re} + jz_{im} &= (x_{re}y_{re} - x_{im}y_{im}) + j(x_{im}y_{re} + x_{re}y_{im}) \end{aligned}$$

Nach dem Ausmultiplizieren wird schnell sichtbar, dass bei dieser Art der Realisierung insgesamt vier Multiplikationen und zwei Additionen bzw. Subtraktionen benötigt werden. Multiplikationen sind jedoch deutlich aufwändiger und ressourcenintensiver als Additionen, so dass eine Lösung mit weniger Multiplikationen hilfreich wäre. Betrachtet man die obige Formel noch einmal wird klar, dass durch Umstellen eine Multiplikation eingespart werden kann:

$$\begin{aligned} z_{re} &= (y_{re} - y_{im}) * x_{im} - (x_{re} - x_{im}) * y_{re} \\ z_{im} &= (y_{re} + y_{im}) * x_{re} - (x_{re} - x_{im}) * y_{re} \end{aligned}$$

Zwar finden sich immer noch vier Multiplikationen in obiger Formel, jedoch ist der zweite Teil beider Gleichungen identisch und muss somit nur einmal berechnet werden. Insgesamt ergeben sich in dieser Variante fünf Additionen bzw. Subtraktionen sowie drei Multiplikationen.

Zur Realisierung der eigentlichen Multiplikation wurde anfangs ausschließlich auf das Synthesetool vertraut. Die Ergebnisse waren jedoch deutlich schlechter als zunächst erwartet. Nach einigen vergeblichen Versuchen durch Modifikation der Tooleinstellungen zu besseren Resultaten zu gelangen wurde schließlich beschlossen, einen Konstantenmultiplizierer basierend auf einfacheren Operationen wie Additionen und Subtraktionen zu realisieren.

Die Funktionsweise der Konstantenmultiplikation lässt sich am Besten am Ablauf der schriftlichen Multiplikation erklären. Hierbei wird der variable Operand nacheinander mit allen Stellen einer Konstante gewichtet und das Ergebnis ergibt sich aus der Summe der einzelnen Zwischenergebnisse. Dieser Ablauf ist sowohl im dezimalen wie auch im binären Zahlensystem identisch. Im binären System existieren für jede Stelle der

## 5. Realisierung der Signalverarbeitung

---

Konstante jedoch nur die Werte 0 oder 1 und die Gewichtung des Operanden lässt sich sehr einfach, bzw. in der hier ausgeführten parallelen Struktur völlig ressourcenfrei, durch eine Verschiebeoperation realisieren. Daher ist es ausreichend, den zu gewichtenden Operanden an allen 1er Stellen der Konstante entsprechend der Stellenwertigkeit zu verschieben und aufzuaddieren. Eine weitere Möglichkeit zur Optimierung ergibt sich, wenn mehrere aufeinanderfolgende 1sen der Konstante zusammenfasst werden. Anstatt alle verschobenen Varianten des Operanden für alle diese Stellen zu addieren ist es sinnvoller, die Differenz aus der höchst- und niederwertigsten Operandenverschiebung zu berechnen und der Gesamtsumme hinzuzuaddieren.

Ein Beispiel illustriert dieses Vorgehen: Es soll der variable Operand  $10101$  mit den Konstanten  $01110$  multipliziert werden. Das ist mit insgesamt zwei Addierern möglich, welche die verschobenen Werte des Operanden summieren:

$$101010_b + 1010100_b + 10101000_b = 100100110_b$$

Alternativ kann das gleiche Ergebnis auch mit nur einem Subtrahierer berechnet werden:

$$101010000_b - 101010_b = 100100110_b$$

wobei 50% Ressourcen eingespart werden. Auf diese Weise können ab einer Sequenzlänge von drei 1sen im Wert der Konstanten Rechenoperationen eingespart werden. Abbildung 5.2 zeigt die Struktur des entwickelten Designs.

Die verschobenen Operanden und die erste Stufe ihrer Verknüpfungen (additiv oder subtraktiv) werden während der Synthese, basierend auf den jeweiligen eingesetzten Konstanten, ermittelt und ein entsprechendes Schaltnetz wird vollautomatisch erzeugt. Die Summation der Ergebnisse dieser ersten Stufe zum Endergebnis erfolgt schließlich mit Hilfe eines gepipelineten Addierbaums. Dieser wird ebenfalls dynamisch während der Synthese generiert und besitzt exakt halb so viele Eingänge wie die Bitbreite der Konstante. So existieren auch für den ungünstigsten Fall eines 01 Musters der Konstante genügend Eingänge.

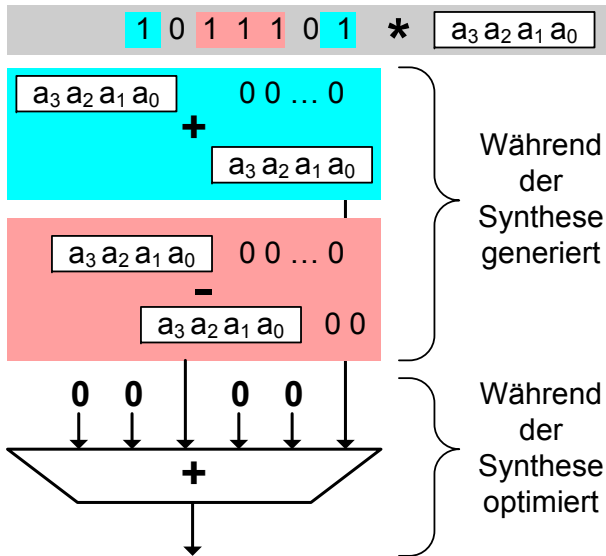


Abbildung 5.2.: Konstantenmultiplikation auf Basis von Addition und Subtraktion

Die für die konkrete Realisierung der ersten Stufe nicht benötigten Eingänge des Addierbaums werden konstant auf Null gesetzt und die dazu benötigten Ressourcen sollten während der Optimierungsphase der Synthese vollautomatisch entfernt werden. Es stellte ich jedoch heraus, dass diese Optimierung zunächst lediglich mit dem Synthesetool *Synplify* von Synopsys zufriedenstellend funktionierte. Das Tool *XST* von Xilinx in einer 13.x Version hingegen erzeugte trotz vieler getesteter Einstellungen immer wieder ein sehr ineffizientes Logikdesign für den Addierbaum. Aus diesem Grund wurde zur Generierung der Netzlisten der hier vorgestellten FFT *Synplify* eingesetzt. Mittlerweile ist es jedoch auch mit *XST* in der Version 14.4 möglich, eine effiziente Netzliste zu generieren. Dieses doch recht einfache Beispiel zeigt deutlich, dass es immer sinnvoll ist die Syntheseergebnisse zu hinterfragen und kritische Operationen bei Bedarf manuell umzusetzen. Tabelle 5.3 fasst die Syntheseergebnisse der Konstantenmultiplikation zusammen.

## 5. Realisierung der Signalverarbeitung

---

Parameter	XST (14.4)	Synplify (J-2015.03-SP1)
Anzahl Register	487	536
Anzahl LUTs	496	352
Maximale Taktfrequenz	555.001 MHz	739.8 MHz

Tabelle 5.3.: Syntheseergebnisse der Konstantenmultiplikation  
(15 Bit breite Eingangsvektoren)

### 5.2.3. Aufbau der FFT

Nachdem die beiden Grundstrukturen zur Berechnung der *Radix-4 Butterflies* sowie der Konstantenmultiplikationen nun bekannt sind, kann der FFT Algorithmus zusammengesetzt werden. Wie bereits in Kapitel 2.2.2 erläutert wurde, existieren grundsätzlich zwei Arten den Algorithmus umzusetzen. Die Varianten DIT und DIF. Um eine Entscheidung zu treffen, welche Variante sich besser zur Realisierung einer ressourcenschonenden und performanten FFT Implementierung eignet, ist es hilfreich noch einmal die ermittelten Performanzdaten der beiden Grundstrukturen zu betrachten. Für beide Strukturen ergibt sich eine Post-Synthese Geschwindigkeit von über 500 MHz. Das ist nicht verwunderlich, denn die einzige kombinatorische Logik, die in beiden Strukturen eingesetzt wird, sind einfache Dual-Input Additionen bzw. Subtraktionen. Bedenkt man, dass eine Transformationsfrequenz von 97.65625 MHz für die spätere Funktion ausreichend ist erkennt man, dass zur Berechnung einer FFT durchaus vier Takte aufgewendet werden können. Die Berechnung kann also (soweit von ihrer Struktur her einfach möglich) vierfach gemultiplext werden. An dieser Stelle macht es jedoch einen großen Unterschied, ob eine DIT oder eine DIF Struktur zum Einsatz kommt. Um das zu verstehen, sollten die beiden Abbildungen 5.3 und 5.4 einmal genauer betrachtet werden. Es handelt sich zur vereinfachten Darstellung lediglich um eine 8 Punkt FFT in *Radix-2* Ausführung. Das Prinzip gilt aber ebenso für *Radix-4* basierte Realisierungen mit einer höheren Anzahl an Punkten.

Es ist leicht zu erkennen, dass in den Bereichen, in denen die Berechnungen in einzelne, unabhängige Bereiche unterteilt werden können, auch die konstanten Faktoren der Multiplikation in jedem Bereich identisch sind. Das ist für die Variante DIT für alle Stufen außer der Letzten der Fall, und

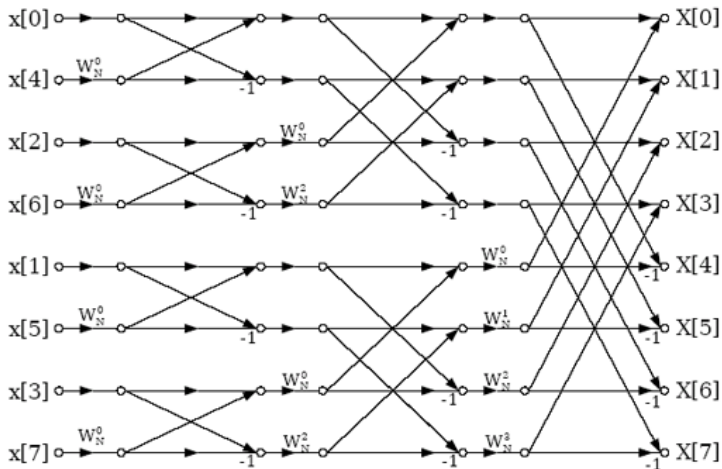


Abbildung 5.3.: DIT Radix-2 8 Punkt FFT

für die Variante DIF für alle Stufen außer der Ersten. Im Fall einer *Radix-4* Struktur ergeben sich in allen Stufen außer der Ersten bzw. Letzten sogar vier identische Bereiche. Diese Struktur eignet sich daher besonders gut für ein vierfaches Multiplexen, da alle Stufen (außer der Letzten bzw. Ersten) identisch arbeiten. Abbildung 5.5 stellt die identischen Bereiche einer *Radix-4* basierten DIF FFT in unterschiedlichen Farben dar.

Aufgrund des Designs der Konstantenmultiplikation ist diese Tatsache besonders wichtig, da somit auch immer die gleichen Konstanten zum Einsatz kommen. Bedenkt man nun noch, dass die Bitbreite der in der Berechnung involvierten Signale zum Ende der Berechnung hin ansteigt, wird klar, warum sich eine DIF Struktur besser zur Realisierung eignet: Die einzige Stufe, die nicht gemultiplext werden kann ist hier auch gleichzeitig die erste Stufe und damit diejenige, mit den kleinsten Bitbreiten der Signale. Abbildung 5.6 zeigt die Struktur der letztendlich realisierten, gemultiplexten FFT.

## 5. Realisierung der Signalverarbeitung

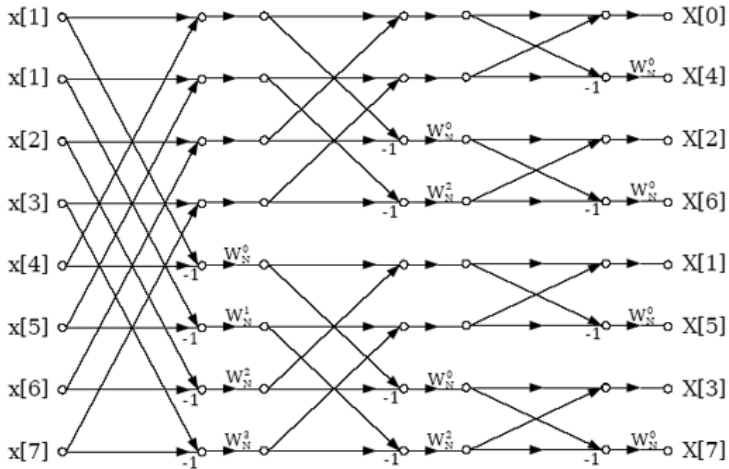


Abbildung 5.4.: DIF Radix-2 8 Punkt FFT

Tabelle 5.4 fasst die Synthesergebnisse der vollständig realisierten 256 Punkte FFT zusammen.

Parameter	XST (14.4)	Synplify (J-2015.03-SP1)
Anzahl Register	174525	162494
Anzahl LUTs	163336	127127
Maximale Taktfrequenz	559.660 MHz	852.1 MHz

Tabelle 5.4.: Synthesergebnisse der 256 Punkte FFT

### 5.3. Zeit- und Frequenzsynchronisation

Neben der Berechnung der FFTs an unterschiedlichen Stellen des Kommunikationssystems erfordert auch die Synchronisation des Downlinks am



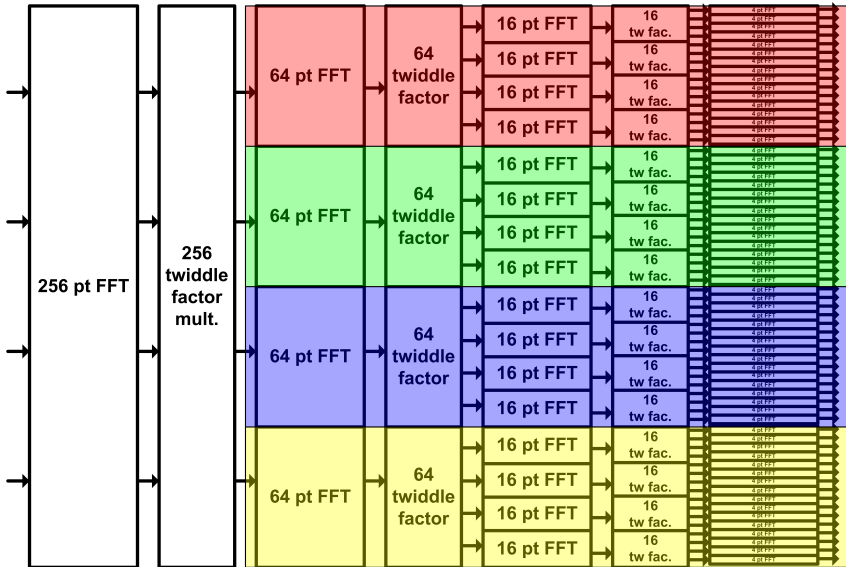


Abbildung 5.5.: 256 Punkt FFT in paralleler Ausführung

ONU verhältnismäßig großen Rechenaufwand. Glücklicherweise besteht dieser Aufwand aufgrund der optimierten Synchronisationsstruktur (vgl. Abschnitt 4.3.1) lediglich beim ONU Empfänger, welcher im Vergleich zum OLT eine deutlich geringere Signalbandbreite verarbeitet.

Zur Synchronisation des internen *Framecounters* mit dem eintreffenden Downlinksignal muss eine ONU den Beginn der Synchronisationssequenz zuverlässig erkennen. Das Mittel der Wahl dazu ist die Verwendung einer Autokorrelation, da so Signalveränderungen des Downlinks durch die Übertragungstrecke weitgehend ohne Einfluss bleiben. Das Maximum der Amplitude des komplexen Korrelationsergebnisses liefert den exakten Empfangszeitpunkt einer Synchronisationssequenz, während die Phasenlage den Frequenzoffset zwischen den Lokalschillatoren von OLT und ONU angibt (vgl. auch 4.2.4).

In den folgenden Unterabschnitten wird die Realisierung der einzelnen dabei zum Einsatz kommenden Verarbeitungsblöcke beschrieben.

## 5. Realisierung der Signalverarbeitung

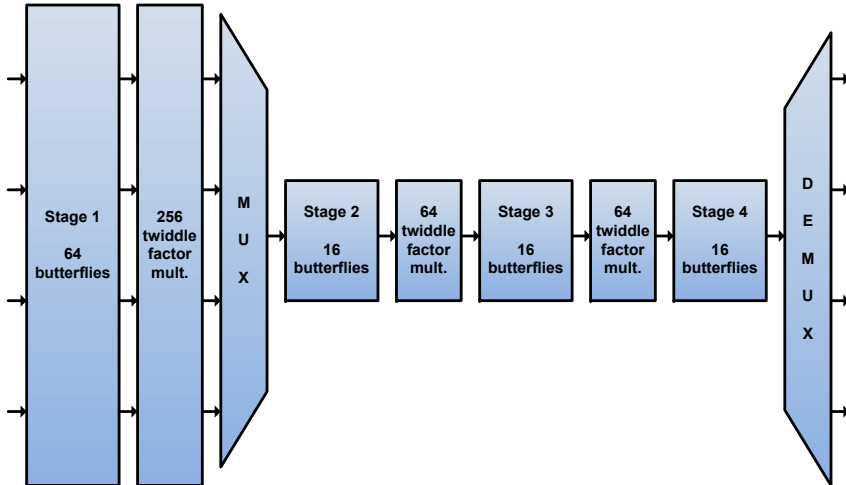


Abbildung 5.6.: 256 Punkt FFT in Multiplexausführung

### 5.3.1. Autokorrelation

Ein wesentliches Problem bei der Berechnung der Autokorrelation in diesem Kommunikationssystem ist, dass die Berechnung nicht wie bei einer FFT in abgeschlossenen Paketen stattfinden kann, sondern über einen kontinuierlich eintreffenden Datenstrom erfolgt. Dabei muss beachtet werden, dass an einer ONU in jedem Takt 16 komplexe Samples der ADCs eintreffen. Diese müssen sowohl untereinander als auch mit den Samples der vorherigen sowie nachfolgenden Takte verknüpft werden.

Die Berechnung der diskreten Autokorrelation erfolgt, indem das empfangene Signal mit einer verzögerten Version seiner selbst multipliziert und für einen gewissen Zeitbereich aufsummiert wird. Da der Aufbau der Synchronisationssequenz bekannt ist, muss die Berechnung nur für exakt eine Signalverzögerung erfolgen. Dabei handelt es sich um 5 Symbolängen (200 Samples), also den Abstand zwischen den beiden identischen Teilen, aus denen die Synchronisationssequenz aufgebaut ist. Nur in diesem Abstand kann eine signifikante Korrelation aus der Synchronisationssequenz resultieren. Der Zeitbereich für den die Multiplikationsergebnisse

addiert werden, im Folgenden auch als Korrelationsfenster bezeichnet, ergibt sich aus der Dauer einer Synchronisationssequenz (400 Samples). So ist sichergestellt, dass für den Fall, dass sich die gesamte Sequenz im Korrelationsfenster befindet, alle Samples zur Detektion beitragen.

Da sich das Korrelationsfenster mit jedem eintreffenden Sample verschiebt, muss auch die Berechnung der Autokorrelation theoretisch jedes Mal wiederholt werden. Praktisch ist aber leicht ersichtlich, dass sich die Korrelationssumme mit jedem empfangenen Sample nur um zwei Summanden verändert. Zum einen muss das Multiplikationsergebnis des neu eingetroffenen Sample mit dem verzögerten Sample hinzuaddiert werden. Zum anderen muss das älteste Multiplikationsergebnis aus der Korrelationssumme entfernt werden. Dieses Vorgehen reduziert den Rechenaufwand signifikant.

Da in dem hier vorgestellten System pro Takt 16 neue Samples eintreffen, werden in jedem Takt auch 16 Multiplikationsergebnisse zur Summe hinzuaddiert und gleichzeitig 16 Alte entfernt. Weiterhin muss beachtet werden, dass für die nachfolgende Auswertung nicht nur die Korrelationssumme nach der Addition der 16 neu eingetroffenen Samples benötigt wird, sondern auch alle Zwischenergebnisse. Entsprechend müssen 16 Summen parallel verwaltet werden, wobei jede eine andere Anzahl von neuen Multiplikationsergebnissen pro Takt aufnimmt. Das Ausgangssignal des Autokorrelators besteht entsprechend, wie auch das Eingangssignal, aus 16 komplexen Werten pro Takt.

#### 5.3.2. Spitzendetektion

Nachdem die Autokorrelation berechnet wurde, muss zur korrekten Detektion des Empfangszeitpunkts der Synchronisationssequenz die Spitze der Amplitude des Korrelationssignals erkannt werden. Dazu werden zunächst 16 parallel arbeitende CORDIC Algorithmen eingesetzt, die aus der kartesischen Repräsentation des Korrelationsergebnisses Amplitude und Phase berechnen.

Da die Amplitude neben der Ähnlichkeit des Signals zu der verzögerten Version seiner selbst auch von der empfangenen Signalenergie abhängig ist, wird zusätzlich noch die Signalenergie benötigt. Deren Berechnung

## 5. Realisierung der Signalverarbeitung

---

erfolgt analog zur Berechnung der Autokorrelation, jedoch mit einer Signalverzögerung von Null. D.h. das Signal ist sich maximal ähnlich.

Im Idealfall würde nun das Korrelationssignal mit Hilfe des Energiesignals normalisiert. Um den Aufwand einer Division einzusparen kommt stattdessen jedoch eine FSM zum Einsatz, die eine Spitzendetektion des Korrelationssignals durchführt und dabei gleichzeitig das Energiesignal über Verhältnissfaktoren berücksichtigt.

Sobald eine gültige Spitze erkannt wurde, wird auch die Phase des Korrelationssignals übernommen und zusammen mit einem Puls, der die zeitliche Lage der Spitze innerhalb der 16 parallel eintreffenden Samples signalisiert, ausgegeben. Damit ist das Synchronisationsmodul komplett. Die ausgegebenen Signale werden vom *Framecounter* verarbeitet und genutzt, um die nachfolgenden OFDM Symbole für die Weiterverarbeitung mittig aus dem eintreffenden Samplestrom zu entnehmen sowie die korrekte Rahmenzeit zu generieren.

### 5.3.3. CORDIC Algorithmus

An mehreren Stellen im Signalverarbeitungssystem wird ein Wechsel einer komplexen Zahlendarstellung von Kartesischen- in Polarkoordinaten benötigt. Dies ist beispielsweise immer dann der Fall, wenn der Betrag oder die Phase eines Signals ermittelt werden muss, wie beispielsweise bei der Spitzendetektion, welche im letzten Abschnitt beschrieben wurde. Die klassische Realisierung mit Hilfe von Sinus oder Cosinus Tabellen ist für eine Umsetzung im entwickelten Signalverarbeitungssystem auf Grund der großen Bitbreiten und vielfach parallelen Prozessierung von Signalen schlecht geeignet. Die erforderlichen Tabellen bei einer Signalbreite von bis zu 18 Bit, welche sich nach der Berechnung der FFT am ONU einstellt, würden zu viele Ressourcen einnehmen. Stattdessen kommt mit dem zweidimensionalen CORDIC Algorithmus ein Verfahren zum Einsatz, welches bereits seit den 1960er Jahren in vielen Rechen-einheiten Verwendung findet. Damit der iterative Algorithmus auch in dieser Anwendung die notwendige Performanz erreicht, wurde er in einer gepipelinten Struktur umgesetzt, so dass pro Takt ein Eingangswert verarbeitet werden kann.

### 5.3.4. Numerically controlled oscillator

Ein weiteres Funktionsmodul, welches immer wieder an verschiedenen Stellen des Systems Verwendung findet, ist ein NCO. Dabei handelt es sich praktisch um den digitalen Kern einer DDS. Prinzipiell ist der Aufbau recht einfach: Ein Phasenakkumulator wird mit jedem Takt um einen gewünschten Wert inkrementiert (bzw. dekrementiert). Anschließend werden die oberen Bits des Akkumulatorregisters, welche die aktuelle Phase in der gewünschten Auflösung repräsentieren, mit Hilfe einer Sinus- bzw. Kosinustabelle in ein Ausgangssignal umgewandelt. Da an dieser Stelle die geforderten Auflösungen nicht ganz so hoch sind wie bei der Anwendung des CORDIC Algorithmus (maximal 12 Bit), können hier Tabellen zum Einsatz kommen.

Typischerweise wird sowohl ein Sinus als auch ein Kosinus Ausgangssignal benötigt, da häufig komplexe Signale kontinuierlich in ihrer Phasenlage gedreht werden müssen, wie beispielsweise zur Korrektur des Lokaloszillatoroffsets. Zur Reduktion des Speicherbedarfs der Tabellen wird lediglich der erste Quadrant einer Sinustabelle in einem Read Only Memory (ROM) abgelegt. Die weiteren Quadranten können durch entsprechende Invertierung des Winkels bzw. des Ausgangssignals sehr einfach abgebildet werden.

### 5.3.5. Equalizer

Wie in Abschnitt 4.3.1 und 4.3.2 erläutert wurde, muss beim ONU Empfänger bzw. ONU Transmitter direkt nach bzw. vor der Berechnung der (I)FFT im Frequenzbereich die Amplitude und Phase jedes einzelnen Subträgers korrigiert werden. Das ist notwendig, da die Signalleistung bzw. die Phasenlage der einzelnen Träger abhängig von der Entfernung zwischen OLT und ONU variieren.

Die Korrektur beider Parameter kann dabei durch die Multiplikation eines komplexen Entzerrungsfaktors erfolgen. Anders als bei der Realisierung der FFT sind diese Faktoren jedoch über die Laufzeit des Systems variabel, da auch Temperaturschwankungen und weitere Effekte Einfluss insbesondere auf die Phasenlage der Subträger haben. Aus diesem Grund kann an dieser Stelle nicht auf eine vereinfachte Konstantenmultiplikati-

## 5. Realisierung der Signalverarbeitung

---

on, basierend auf Additionen, zurückgegriffen werden, sondern es wird eine eine echte, komplexe Multiplikation zwischen zwei variablen Werten benötigt.

Die Berechnung der Entzerrungsfaktoren erfolgt aus den Phasenreferenzdaten (vgl. 4.2). Für den Downlink ist bekannt, dass alle während der Phasenreferenzsequenz aktiven Träger mit einer Amplitude von 1.0 und einer Phasenlage von 0 Grad ausgesendet wurden. Entsprechend kann der Entzerrungsfaktor  $eq$  zur Kompensation der Kanaleinflüsse für Subträger  $i$  mit Hilfe einer komplexen Division aus dem Konstanten Wert  $1+0j$  und der empfangenen Phasenreferenz  $ref$  für Subträger  $i$  berechnet werden:

$$eq = \frac{1+0j}{ref}$$

Da während der Übermittlung der Phasenreferenz in jedem gesendeten Symbol lediglich ein Subträger einer spektralen Gruppe aktiv ist, können die Entzerrungsfaktoren nacheinander berechnet werden und es wird lediglich ein komplexer Dividierer benötigt. Dieser ist jedoch gepipelined aufgebaut, so dass er pro Takt ein Divisionsergebnis liefern kann.

Die Berechnung der Entzerrungsfaktoren für die Vorentzerrung des Uplinksignals im ONU Sender erfolgt im Prozessorsystem des OLT Empfängers aus den dort aufgenommenen Phasenreferenzdaten. Anschließend werden die berechneten Faktoren mit Hilfe des Downlinks an die entsprechende ONU übermittelt und dort angewandt.

### 5.3.6. Frequenzoffsettracking

Da die Oszillatoren der ONUs aufgrund vieler Effekte wie z.B. Fertigungstoleranzen oder Temperaturschwankungen niemals synchron zu denen des OLT arbeiten, ist ein Frequenzoffsettracking mit nachfolgender Korrektur des Frequenz- bzw. Phasenfehlers unabdingbar für den korrekten Empfang phasenmodulierter Daten. Wie bereits in Abschnitt 4.3.1 beschrieben, erfolgt die Korrektur der Oszillatoren zweistufig. Zunächst aufgrund der Messungen im Synchronisationsmodul und nach erfolgter Synchronisation mit Hilfe der Pilottöne. Während die vom Synchronisationsmodul gemessenen Frequenzfehler direkt dem Lokal- bzw.

Samplingszillator zur Korrektur zugeführt werden, müssen die Werte der Pilotöne zunächst vorverarbeitet werden. In einem ersten Schritt extrahiert ein CORDIC Algorithmus aus der kartesischen Darstellung der beiden Pilotöne die momentane Phasenlage. Im Anschluss daran kann durch einfache Summen- bzw. Differenzbildung der beiden Pilotonphasen der gemeinsame bzw. differentielle Phasenfehler des eintreffenden Signals ermittelt werden. Da die gemessenen Phasenfehler je nach Empfangsqualität jedoch von einem mehr oder weniger starken Rauschen überlagert sind, werden die Werte nicht direkt an die Oszillatoren weitergegeben. Stattdessen ist für jeden Phasenfehler ein PID Regler vorgesehen, welcher eine sanfte und ruckfreie Ausregelung des Fehlers ermöglicht.

### 5.3.7. Synchronisation des Uplinks

Wie bereits in Abschnitt 4.2.4 beschrieben, erfordert die Synchronisation der einzelnen ONUs lediglich eine exakte zeitliche Ausrichtung des Rahmenbeginns bei den ONU Transmittern. Frequenzoffsets von Lokal- und Sampleszillator müssen nicht mehr korrigiert werden, da dieselben Oszillatoren wie beim Downlink verwendet werden und deren Frequenzoffsets bereits durch die Downlinksynchronisation korrigiert werden. Die von einer ONU ausgesendete Synchronisationssequenz besteht aus zwei Perioden einer Sinusschwingung und der Rahmenbeginn des Uplinks kann an jedem ONU durch einen *Timing Advance* genannten Parameter sample-genau im Zeitbereich verschoben werden.

Die Erzeugung der Sinusschwingung am ONU gestaltet sich verhältnismäßig einfach. Durch die ausschließliche Modulation des physikalischen Subträgerindex +1 erzeugt die im Transmitter enthaltene IFFT eine reine Sinus- bzw. Kosinusschwingung mit einer Periodenlänge, die einer OFDM Symbollänge entspricht. Durch Modulation des Subträgers über zwei Symbolauern entsteht das erwünschte Synchronisationssignal.

Auch der Empfang des Signals am OLT gestaltet sich einfach und ressourcenschonend, was auch der Grund für dieses abgewandelte Synchronisationsverfahren im Uplink ist. Durch eine komplexe Kreuzkorrelation des Empfangssignals mit einer Referenzschwingung lässt sich das Vorhandensein einer Synchronisationssequenz im Empfangsdatenstrom und dessen Phasenlage genau bestimmen. Im Gegensatz zur Autokorrelation

(wie sie beim Downlinkempfänger zum Einsatz kommt) muss nicht mit jedem eintreffenden Sample eine neue Berechnung inklusive Auswertung erfolgen, sondern es genügt eine Berechnung pro empfangenem Symbol. Im Falle des OLT entspricht dies 320 Samples und generiert damit bedeutend weniger Rechenaufwand.

Der durch die Synchronisationseinheit des OLT ermittelte Symbolindex und die Phasenverschiebung des empfangenen Signals zum Referenzsignal erlauben schließlich die präzise Bestimmung des *Timing Advance* Parameters für die betreffende ONU.

Der Synchronisationssequenz folgt auch im Uplink eine Phasenreferenz, welche zur Korrektur der Amplituden- und Phasenfehler der einzelnen Subträger durch die Pre-Equalizer der ONUs dient. Da im Uplink jedoch keine Pilottöne zum Einsatz kommen, müssen die Korrekturwerte für die Equalizer nicht sofort bereit stehen. Daher wird diese Sequenz lediglich durch die Empfangshardware im OLT aufgezeichnet und anschließend in Software ausgewertet und die Ergebnisse zu den ONUs gesandt.

### 5.4. Modulatoren und Demodulatoren

Ein wesentlicher Teil des Signalverarbeitungssystems, der bisher nur wenig Beachtung fand, ist die Umsetzung des digitalen Datenstroms in komplexe Symbole mit Hilfe eines Modulators bzw. der umgekehrte Vorgang, die Zuordnung eindeutiger Bitkonstellationen zu empfangenen Symbolen in einem Demodulator.

Durch viele, zum Teil sehr aufwändigen Prozessierungsschritte wie beispielsweise der Berechnung von FFTs, Zeit- und Frequenzsynchronisation oder Entzerrung, werden die analogen Signale des physikalischen Kommunikationskanals im Empfänger aufbereitet bzw. vor dem Senden vorbereitet. Das Ziel dieser aufwändigen Verarbeitung ist die Bildung eines stabilen Kanals für jeden Subträger, der die parallele, amplituden- und phasenstabile Übermittlung von komplexen Symbolen erlaubt. Trotz der digitalen Natur der komplexen Symbole kann deren Übermittlung unter Berücksichtigung der Quantisierung der Symbolkoordinaten mit den gewählten Bitbreiten als eine Art kontinuierliche Übertragung verstanden werden. Kanaleinflüsse wie Rauschen oder Frequenzfehler machen sich



in Form von mehr oder weniger starken Abweichungen der empfangenen Symbolkoordinaten von ihrer idealen Position bemerkbar, was sich an Empfangsdaten mit Hilfe von Konstellationsdiagrammen sehr schön beobachten lässt.

Die Aufgabe eines Modulators bzw. Demodulators ist es nun, die mehr oder weniger kontinuierlich übermittelte Amplituden- bzw. Phaseninformation für die Übertragung digitaler Informationen zu nutzen. Damit ähnelt ihre Aufgabe und Funktionsweise ein wenig der von DA bzw. AD Wandlern, mit dem Unterschied, dass die Konversion in diesem Fall nicht zwischen einem analogen und einem digitalen Signal erfolgt, sondern zwischen digitalen Signalen feiner Quantisierung und digitalen Signalen grober Quantisierung.

Die Realisierung eines Modulators ist tatsächlich recht trivial. Er besteht im Wesentlichen aus einer Look-Up Tabelle, in der für jede mögliche Konstellation die idealen kartesischen Koordinaten des zugehörigen komplexen Symbols hinterlegt sind. Im Betrieb wird die Tabelle einfach mit dem gewählten Bitvektor adressiert und das Symbol kann sofort ausgelesen werden. Je nach Modulationsart (BPSK, PSK, QAM, ...) können verschiedene Tabellen bzw. Tabellenbereiche verwendet werden.

Die Realisierung eines Demodulators kann hingegen deutlich aufwändiger ausfallen. Jedem empfangenen Symbol muss der Bitvektor zugeordnet werden, der am wahrscheinlichsten den Ursprung des Symbols darstellt. Das ist umso aufwändiger, da die empfangenen Symbole in der zweidimensionalen, komplexen Ebene liegen. Für eine mathematisch korrekte Realisierung eines Demodulators müsste für jedes empfangene Symbol der euklidische Abstand zu allen idealen Konstellationspunkten berechnet und anschließend der Punkt mit dem minimalen Abstand ausgewählt werden. Es sollte jedoch klar sein, dass dieser Rechenaufwand insbesondere beim OLT Empfänger mit bis zu 52 parallel eintreffenden Symbolen pro Takt zu aufwändig für dieses System wäre. Stattdessen wird die Symmetrie der in diesem System gewählten Modulationsarten ausgenutzt. Um jeden idealen Konstellationspunkt wird ein rechteckiger Bereich definiert, der diesem Punkt zugeordnet ist. Auf diese Weise kann der Real- bzw. Imaginärteil der kartesischen Repräsentation des empfangenen Symbols mit Hilfe einfacher Vergleichsoperationen in einige wenige horizontale bzw. vertikale Bereiche unterteilt werden, die im

## 5. Realisierung der Signalverarbeitung

---

Folgenden sehr einfach einem bestimmten Bitvektor zugeordnet werden können. In Abbildung 5.7 sind diese Bereiche für den Fall einer QAM-16 Konstellation unterschiedlich eingefärbt.

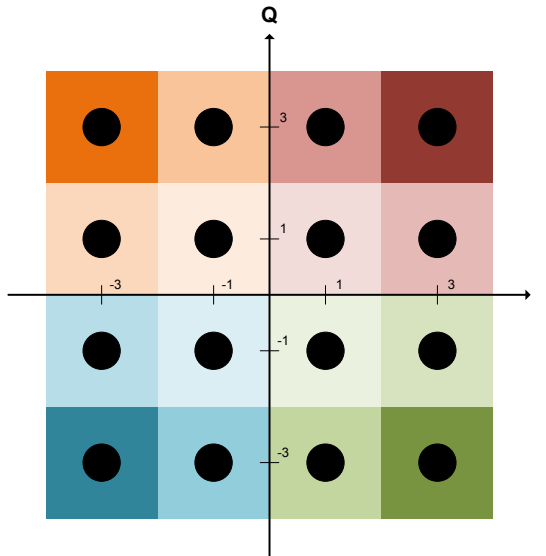


Abbildung 5.7.: QAM-16 Konstellationsbereiche

Eine Ausnahme bildet die 8-PSK Modulation. Hierbei ist die Amplitude des Symbols konstant und die übermittelte Information wird ausschließlich in der Phase kodiert. Zur Demodulation kommt ein CORDIC Algorithmus zum Einsatz, welcher die Phasenlage des empfangenen Symbols extrahiert. Anschließend kann, ähnlich wie oben dargestellt, mit Hilfe von Vergleichsoperationen der korrekte Bitvektor identifiziert werden. Aufgrund der CORDIC Berechnung ist dieses Verfahren deutlich aufwändiger zu realisieren. Es macht daher Sinn, nicht alle spektralen Gruppen am OLT mit der Fähigkeit zur Demodulation von PSK modulierten Subträgern auszustatten. Das Bandbreitenmanagement muss in diesem Fall dafür Sorge tragen, dass ein Teilnehmer, für den eine PSK Modulationsart in einer gegebenen Situation ideal ist, der richtigen spektralen Gruppe zugeordnet wird.

Neben dem ermittelten Bitvektor gibt der in dieser Arbeit aufgebaute Demodulator zusätzlich noch den kartesischen Abstand zwischen tatsächlich empfangenem Symbol und dem nächsten idealen Punkt im Konstellationsdiagramm aus, der durch eine einfache Subtraktion der beiden Werte gebildet werden kann. Diese Information wird zur Zeit nicht weiter ausgewertet, könnte aber in Zukunft bei der Verwendung einer Soft-Decision FEC hilfreich sein.

## 5.5. Paketdatenverarbeitung

Die in Abbildung 4.18 dargestellte *MAC Layer* bildet die Paketdatenverarbeitung dieses Kommunikationssystems. Ihre Realisierung konnte in weiten Teilen ohne besondere Schwierigkeiten erfolgen, da der Datendurchsatz im Verhältnis zu den Signalverarbeitungsblöcken relativ gering ausfällt. Bei voller Auslastung der 10G Ethernet Schnittstelle ergibt sich ein Datendurchsatz von lediglich 64 Bit bei 156 MHz. Allerdings ist die gewählte Realisierung, wie bereits in Abschnitt 4.3.3 erläutert, nur in der Lage maximal zwei ONUs mit Datenströmen zu versorgen. Eine flexiblere Realisierung für eine nahezu beliebige Anzahl von ONUs würde voraussichtlich deutlich mehr Ressourcen erfordern. Aufgrund des Fokus dieser Arbeit auf die OFDM Signalverarbeitungskette genießt eine flexiblere Lösung hier jedoch nicht die höchste Priorität.

Bis auf das *10G Ethernet MAC* Modul und die FEC En- sowie Decoder wurden auch hier sämtliche Komponenten speziell für dieser Arbeit entwickelt. Eine der Hauptschwierigkeiten lag dabei im Design der Module für die Subträger Verteilung bzw. Kombination. Ihre Aufgabe besteht in der Bildung eines Interfaces zwischen einem 64 Bit breiten Datenstrom und den einzelnen Modulatoren bzw. Demodulatoren der OFDM Sender bzw. Empfänger. Der Datenstrom mit fester Bitbreite muss je nach Konfiguration der Subträgermodulation sehr unterschiedlich auf die einzelnen Subträger aufgeteilt werden, wodurch komplexe und große Multiplexerstrukturen entstehen. Erschwerend kommt noch hinzu, dass die Module bei der hohen Taktfrequenz des OFDM Systems arbeiten, d.h. 195 MHz im Fall einer ONU bzw. 390 MHz für das OLT.

## 5. Realisierung der Signalverarbeitung

---

Das 10G Ethernet MAC sowie die Reed-Solomon FEC Module stammen von *Opencores* [30], [31]. Bedingt durch das Funktionsprinzip von FEC Algorithmen benötigt vor allem der Decoder sehr viele Ressourcen obwohl in dieser Arbeit lediglich eine *Hard Decision* Variante der FEC zum Einsatz kommt. Die originale Implementierung von *Opencores* arbeitet lediglich mit einer Datenbitbreite von 8 Bit. Wie auch bei der Realisierung der FFT (siehe Abschnitt 5.2) existieren grundsätzlich zwei Varianten um die 64 Bit breiten Datenvektoren der MAC Layer zu verarbeiten. Zum einen kann die innere Struktur so erweitert werden, dass 64 Bit direkt verarbeitet werden können. Zum anderen können acht Instanzen der originalen Implementierung parallel eingesetzt und der Datenvektor entsprechend aufgeteilt werden. Obwohl die erste Variante ein effizienteres und ressourcenschonenderes Ergebnis erwarten lässt, wurde an dieser Stelle auf Grund des reduzierten Implementierungsaufwandes die zweite Variante gewählt.

Der Realisierungsaufwand der restlichen Module der Paketdatenverarbeitung gestaltet sich verhältnismäßig gering. Zur einfachen und flexiblen Verdrahtung der einzelnen Module wie z.B. Filter, FIFO oder Multiplexer wurde ein universelles Interface für den Paketdatenaustausch realisiert. Es basiert neben dem 64 Bit breiten Datenbus auf einfachen Steuersignalen, welche Beginn, Aktivität und Ende einer Paketübertragung signalisieren. Außerdem existiert ein Signal, welches die Anzahl der gültigen Bytes im letzten Datenwort angibt, da je nach Größe des Ethernetpakets hier auch Werte ungleich acht möglich sind. Auf diese Weise kann die Anordnung der Verarbeitungsmodule sehr flexibel und einfach modifiziert werden, etwa falls mehr als zwei ONUs unterstützt werden sollen oder wenn zusätzliche Priorisierungsebenen für Pakete benötigt werden.

## 5.6. I/O Schnittstellendesign

### 5.6.1. ADC/DAC Anbindung an OLT

Die Anbindung der unterschiedlichen AD bzw. DA Wandler an den FPGA des Mainboards bildet eine der aufwändigsten und komplexesten Teilaufgaben dieser Arbeit. Da es sich um IO Schnittstellen zu externen

Bausteinen handelt, kann die normalerweise verwendete Simulation basierend auf z.B. Modelsim an vielen Stellen nicht angewandt werden. Stattdessen kommt Chipscope sowie ein klassisches Oszilloskop zur Kontrolle der MGLs zum Einsatz. Die hohe Frequenz der Kommunikationslinks im GHz Bereich sowie die umfangreichen und komplexen Konfigurationsmöglichkeiten der GTX Transceiver erschweren die Inbetriebnahme der Schnittstelle zusätzlich.

In diesem Abschnitt wird daher zunächst die Integration der 25 GSa/s Wandler von Micram betrachtet und im nächsten Abschnitt erfolgt eine Übersicht über die Anbindung der langsameren 3.125 GSa/s Wandlermodule, deren Hardwareaufbau im nächsten Kapitel in Abschnitt 6.3 beschrieben wird.

Die Micram Wandler besitzen eine Auflösung von sechs Bit bei einer Abtastfrequenz von 25 GSa/s. Für ihre Anbindung an den FPGA des Mainboards kommen MGLs zum Einsatz, die von GTX Transceivern mit einer Übertragungsrate von 6.25 GBit/s gespeist werden. Jeweils vier dieser MGLs versorgen in einem Micram Wandler über einen Multiplexer eines der sechs Bit. Entsprechend werden für alle sechs Bits gemeinsam 24 MGLs und damit 24 GTX Transceiver benötigt.

Zur Anbindung der Wandler wurde eine Interfacelogik für die Integration ins FPGA entwickelt, welche der Anwenderlogik für jeden Wandler Vektoren zum Zugriff zur Verfügung stellt, die in jedem Takt 64 Samples transportieren. Die Interfacelogik sorgt für den korrekten und parallelen Zugriff auf die GTX Transceiver, für die anfängliche Initialisierung der Wandler sowie die Aufteilung bzw. Zusammenführung der Datenströme der einzelnen Transceiver.

Die Wandler setzen nicht wie üblich einen Linecode zur Synchronisation der einzelnen MGLs ein. Stattdessen wird das Datensignal direkt oder mit einer 127 Bit langen Pseudorandom Binary Sequence (PRBS) Sequenz XOR-verknüpft übermittelt. Zu Beginn müssen daher die Datenströme der 24 MGLs vom FPGA Interface so synchronisiert werden, dass alle sechs zu einem ADC oder DAC Wort gehörigen Bits parallel am Samplevektor der Userlogik anliegen. Da jeder GTX Transceiver auch interne FIFOs besitzt und mit vielen unterschiedlichen Taktdomänen arbeitet, ergeben sich nach jedem Neustart des Systems leicht unterschiedliche Verzögerungsverhältnisse der Transceiver. Außerdem spielen bei Bi-

## 5. Realisierung der Signalverarbeitung

traten im Bereich mehrerer GBit/s auch unterschiedliche Längen der Verbindungsleitungen zwischen FPGA und Wandler eine Rolle.

Zur Synchronisation kommt ein Testmuster basierend auf der bereits zuvor genannten PRBS zum Einsatz. Die Bitsequenz und der Abtastzeitpunkt der einzelnen Bits werden vom FPGA Interface während einer Initialisierungsphase so lange verschoben, bis alle 24 Kanäle synchron arbeiten. Anschließend werden die Nutzdaten mit der PRBS Sequenz XOR verknüpft übermittelt. Gleichzeitig müssen auch die von den einzelnen GTX Transceivern verarbeiteten Bitsequenzen korrekt sortiert und mit dem Samplevektor der Userlogik verbunden werden. Bedenkt man, dass insgesamt 64 Samples pro Takt und Wandler, also 384 Signalleitungen vorhanden sind wird klar, dass auch hier ein hohes Fehlerpotential liegt. Abbildung 5.8 zeigt ein Blockdiagramm des entwickelten FPGA Interfaces.

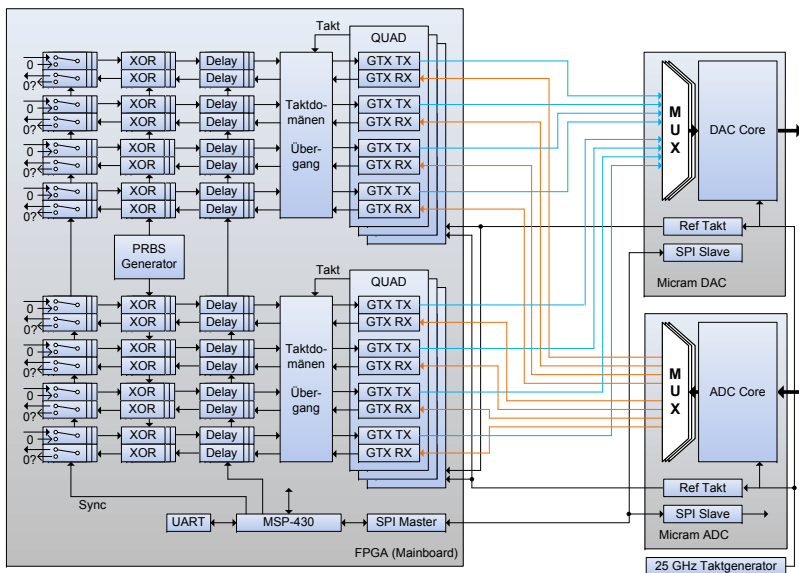


Abbildung 5.8.: Blockdiagramm des FPGA Interfaces für die Hochgeschwindigkeitswandler

Wie der Abbildung entnommen werden kann, müssen neben den oben genannten Punkten noch eine Reihe weiterer Schwierigkeiten beim De-

sign des FPGA Interfaces beachtet werden. Dazu gehört beispielsweise die korrekte Taktversorgung der einzelnen GTX Transceiver. Diese sind jeweils zu viert in sog. *QUADs* angeordnet und teilen sich einen Referenztakteingang. Dieser Referenztakt kann noch von zwei weiteren *QUADs* in direkter Umgebung mitgenutzt werden, so dass sich insgesamt 12 GTX Transceiver einen Referenztakt teilen können. Da für den Zugriff auf einen Wandler 24 Transceiver benötigt werden, muss der Referenztakt des entsprechenden Wandlers mindestens an zwei Stellen zum FPGA geführt werden.

Weiterhin muss beachtet werden, dass auch die Taktversorgung des FPGA Logik Interfaces der GTX Transceiver lediglich für die Hälfte der 24 Transceiver möglich ist. Entsprechend müssen für die Zusammenführung der Signale der 24 eingesetzten Transceiver mehrere Taktdomänen über asynchrone FIFOs gekoppelt werden. Tatsächlich müssen aufgrund der verwendeten IQ Basisbandsignalisierung sogar jeweils die Samplevektoren von zwei ADCs bzw. DACs auf eine Taktdomäne zusammengeführt werden.

Eine weitere Schwierigkeit ergibt sich aus der Tatsache, dass sich jeweils ein AD sowie ein DA Wandler einen GTX Transceiver teilen, da der ADC den Empfangsteil und der DAC den Sendeteil des Transceivers nutzt. Entsprechend verknüpft sind auch hier die Signale der verschiedenen Wandler.

Aufgrund der Komplexität der zu Beginn nach jedem Systemstart durchzuführenden Synchronisation der einzelnen Kanäle kommt ein kleiner Softwareprozessor für die Ablaufsteuerung zum Einsatz. Dabei handelt es sich um einen auf OpenCores verfügbaren Nachbau des bekannten MSP-430 CPU Kerns von Texas Instruments. Zusammen mit dem freien C-Compiler *mspgcc* kann er auf einfache Weise in einer Hochsprache programmiert genutzt werden, um die Bitdelaylines der einzelnen GTX Transceiver mit verschiedenen Werten zu programmieren und das Ergebnis entweder im DA Wandler oder der FPGA Logik zu analysieren. Für die Kommunikation mit den Micram Wandlern kommt ein einfaches, serielles Protokoll zum Einsatz. Auch dieses wird (unterstützt durch ein speziell designtes Hardwaremodul für den MSP-430 Prozessor) von diesem bedient. Das gesamte CPU Subsystem ist aufgrund der relativ moderaten Leistungsfähigkeit des Prozessors recht klein. Es steht mit

dem *Leon-3* Prozessorsystem, welches sowohl in OLT auch auch ONU als Mastercontroller fungiert, über eine asynchrone, serielle Schnittstelle in Verbindung.

### 5.6.2. ADC/DAC Anbindung an ONU

Die Anbindung der langsameren Wandler mit einer Abtastrate von 3.125 GSa/s erfolgt in zwei Schritten. Zunächst muss eine Kommunikationsverbindung zwischen dem FPGA des Mainboards und dem des Wandlermoduls aufgebaut werden. Anschließend wird die Verbindung zwischen dem eigentlichen AD bzw. DA Wandlerchip und dem Modul-FPGA hergestellt und beide Kommunikationskanäle verknüpft. Details zum Aufbau der Wandlermodule werden in Abschnitt 6.3 erläutert.

Für die Verbindung zwischen den beiden FPGAs stehen acht unidirektionale 6.25 GBit/s MGLs basierend auf GTX Transceivern zur Verfügung. Je nach Typ des Wandlermoduls (AD oder DA) und je nach FPGA werden entsprechend die Sende- bzw. Empfangsleitungen der GTX Transceiver miteinander verbunden und der Gesamtdatenstrom der Wandler von 37.5 GBit/s wird auf die einzelnen Links aufgeteilt. Jeder Link wird mit einem 8B10B Linecode betrieben und über spezielle Kommasymbole werden die einzelnen Kanäle synchronisiert und ausgerichtet.

Die Aufteilung des Datenstroms auf die einzelnen Links erfolgt statisch. Da jeder GTX Transceiver in der gewählten Konfiguration 32 Bit pro Takt verarbeitet, wurde für den Gesamtdatenstrom eine Wortbreite von 256 Bit gewählt, welches in 8 Segmente aufgeteilt und mit den einzelnen GTX Transceivern verbunden wird. Da jeder Transceiver mit 6.25 GBit/s und einer 8B10B Kodierung arbeitet, ergibt sich eine Gesamtbandbreite der Schnittstelle von 40 GBit/s. Bei der Einbettung des ADC bzw. DAC Datenstroms von *nur* 37.5 GBit/s kommt es folglich immer wieder zu Leerlaufzuständen. In diesem Fall werden spezielle, im 8B10B Code definierte Kommasymbole in den kontinuierlichen Datenstrom zwischen den FPGAs eingefügt. Diese werden neben der Signalisierung des Leerlaufzustands auch zur Ausrichtung der einzelnen Linkkanäle untereinander ausgenutzt. Aufgrund des komplexen inneren Aufbaus der GTX Transceiver mit verschiedenen Taktdomänen und asynchronen Schaltungsteilen, variiert die Signallaufzeit einzelner Links trotz identischer Referenzakte



nach jedem Neustart des Systems ein wenig. Der Empfänger muss daher die zeitliche Ausrichtung der einzelnen Kanäle zunächst ermitteln und korrigieren, bevor er den 256 Bit breiten Gesamtdatenstrom aus den einzelnen GTX Empfängern wiederherstellen kann. Da die zeitliche Differenz durchaus mehrere zehn Takte betragen kann, erfordert dieser Schritt einen recht breiten Multiplexer, wodurch der Aufbau des Korrekturmoduls bei der erforderlichen Betriebsfrequenz von 195.3125 MHz deutlich erschwert wird.

Die Anbindung der eigentlichen Wandlerchips verlief hingegen relativ problemlos. Die differentiellen IO Leitungen wurden mit entsprechenden FPGA Primitiven angebunden und über einfache DDR Register angesteuert. Da die verwendeten 3.125 GSa/s Wandler wie auch die schnelleren 25 GSa/s Wandler intern jeweils vier digitale Datenleitungen zu einem Bit multiplexen, ergibt sich an der digitalen Schnittstelle der Wandler eine Wortfrequenz von 781.25 MHz. Da diese mit DDR Registern bedient wird, ergeben sich für die FPGA Logik entsprechend acht 12 Bit Wandlerwörter, oder 96 Bits, bei einer Frequenz von 390.625 MHz. Zur Verbindung mit dem GTX Transceiver Interface werden diese 96 Bits pro Takt zunächst auf 192 Bit alle zwei Takte konvertiert und anschließend die Frequenz mit Hilfe eines asynchronen FIFOs ein weiteres Mal auf 195.3125 MHz, die Frequenz der GTX Interfaces, halbiert. Schließlich wird der 192 Bit breite Datenvektor mit dem 256 Bit Vektor des GTX Interface so verknüpft, dass in jedem vierten Takt keine Daten über das Interface versendet werden und entsprechend ein Kommasymbol zur Synchronisation in jedem Link Platz findet.

Auch bei der Anbindung der Wandlermodule bestand eine der Hauptschwierigkeiten wieder darin, eine funktionsfähige Konfiguration der GTX Transceiver zu ermitteln, sowie die Datenlinks mit Hilfe von Chipscope und weiteren Messinstrumenten einzurichten.

Die Abbildungen 5.9 und 5.10 zeigen die Blockdiagramme der ADC bzw. DAC Anbindungen.

Ähnlich den Micram Wandlern besitzen auch die Wandlermodule einige Einstellmöglichkeiten, welche das GTX Interface bzw. auch die AD bzw. DA Wandler Chips direkt betreffen. Aus diesem Grund enthalten die FPGAs aller Wandlermodule eine Slaveimplementierung der seriellen Schnittstelle der Micramwandler. So kann auch hier ein MSP-430 Softcore

## 5. Realisierung der Signalverarbeitung

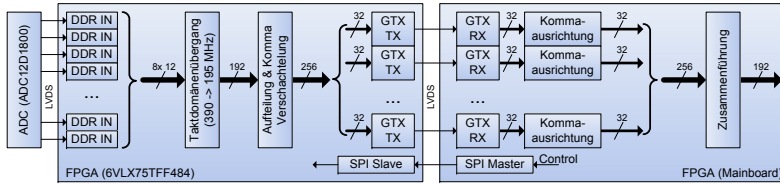


Abbildung 5.9.: Logische Anbindung der ONU ADC Wandlerchips

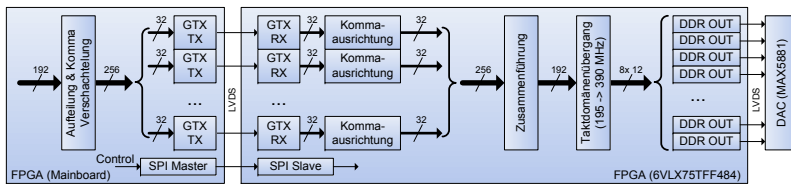


Abbildung 5.10.: Logische Anbindung der ONU DAC Wandlerchips

Prozessorsystem zur Konfiguration, Initialisierung und Überwachung des gesamten Wandler-Subsystems zum Einsatz kommen.

### 5.6.3. Inter-Board Kommunikation

Wie bereits erwähnt, ist am OLT lediglich ein FPGA nicht ausreichend für die Verarbeitung der anfallenden Daten. Aus diesem Grund müssen mehrere FPGA Boards parallel genutzt werden und dazu natürlich auch entsprechend leistungsfähig vernetzt sein. In diesem Abschnitt wird die dazu verwendete Schnittstelle im Detail beschrieben. Es handelt sich dabei um eine leistungsfähige, bidirektionale Kommunikationsverbindung auf Basis von GTX Transceivern. Der zugreifenden FPGA Logik stellt sich das Interface in Form von zwei entgegengesetzt arbeitenden, asynchronen FIFOs dar, welche einen transparenten Datenaustausch zwischen den Logikblöcken unterschiedlicher Bausteine ermöglichen. Somit ist ein universelles Interface gegeben und die Aufteilung der Signalprozessierungslogik kann recht flexibel zwischen in sich geschlossenen Blöcken erfolgen.

Der Zugriff auf die GTH Transceiver geschieht über eine XGMII Schnittstelle. Dabei handelt es sich um das *media independent interface* des 10G Ethernet Standards, der wie auch jeder andere Ethernet Standard ein solches Interface besitzt, um die Anbindung unterschiedlicher physikalische Schichten an die logische Netzwerkschicht zu ermöglichen. Das XGMII Interface ist relativ einfach aufgebaut. Es besteht aus 64 Datenleitungen für die Übermittlung von Nutz- bzw. Kommandodaten, sowie 8 Kontrollleitungen, welche die Art der Information (Nutz- oder Kommandodaten) einzeln für jedes Byte auf den Datenleitungen signalisieren. Die Signalisierung auf dem MGL erfolgt im 64B66B Linecode mit 10.3125 GBit/s, so dass sich eine Nettolinkdatenrate von 10 GBit/s einstellt.

Die grundlegenden Aufgaben der Inter-Kommunikationsschnittstelle können wie folgt zusammengefasst werden:

- Paket Generierung / Paket Auflösung
- Daten Integritätsprüfung
- Automatische Paketwiederholung bei Übertragungsfehlern
- Taktdomänen Übergang

### 5.6.3.1. Schnittstellendesign

Die Nutzdaten bestehen aus einem kontinuierlichen Datenstrom, welcher von der Schnittstelle nahtlos über den MGL transportiert werden muss. Wie jedoch in Abschnitt 2.4.2 erläutert wurde, treten bei jeder Übertragung mittels eines MGL hin und wieder Bitfehler auf. Der kontinuierliche Datenstrom muss daher in einzelne Datenpakete unterteilt werden, um eine Absicherung der Übertragung und Neuanforderung verfälschter Pakete zu ermöglichen. Ausgestattet mit Kontrolldaten und einer CRC Prüfsumme werden diese Pakete über den MGL übertragen. Die Empfangsseite bestätigt dem Sender korrekt empfangene Datenpakete mit Hilfe des Datenlinks der entgegengesetzten Richtung. Falls der Sender in einer festgelegten Zeitspanne keine Bestätigung für einen korrekten Empfang erhält, wiederholt er automatisch die Übertragung des(der) letzten Pakets(Pakete).

Zur Reduktion der Latenz und gleichermaßen Erhöhung des Durchsatzes der Übertragung können mehrere Pakete versendet werden, bevor die

## 5. Realisierung der Signalverarbeitung

---

Übertragung aufgrund fehlender Empfangsbestätigungen gestoppt wird. Um potentielle Übertragungswiederholungen zu ermöglichen, muss der Sender Pufferspeicher bereit stellen, in dem er bereits versendete Pakete aufbewahrt bis deren Empfang bestätigt wird. Grundsätzlich arbeitet der gewählte Ansatz ähnlich wie das Transmission Control Protocol (TCP) Protokoll, welches im Internet breite Anwendung findet. Allerdings kann aufgrund der gewählten MGT Konfiguration keine Vertauschung von Paketen auftreten, so dass auch keine Paketsortierung erforderlich ist. Es sollte an dieser Stelle erwähnt werden, dass auch MGLs sehr zuverlässig sind und Bitfehler nur sehr selten auftreten. Somit ist die zusätzlich benötigte Bandbreite für die Neuübertragung fehlerhafter Pakete ebenfalls äußerst gering. Ohne Datenintegritätsprüfung und Korrektur würde jedoch früher oder später ein Bitfehler die Inter-Kommunikationsschnittstelle oder die Nutzerlogik aufgrund unplausibler Daten zum Absturz bringen, wie in vorhergehenden Experimenten festgestellt wurde. Damit genügend Spielraum für eventuelle Neuübertragungen von Paketen bleibt, sollte die maximale Bandbreite der Schnittstelle niemals voll ausgeschöpft werden.

### 5.6.3.2. Paketformat

Das Paketformat, welches für die Inter-Kommunikationsschnittstelle eingesetzt wird, ist in Abbildung 5.11 dargestellt.

Wie leicht aus der Abbildung zu erkennen ist, wurde die Wortlänge auf 64 Bit festgelegt. Damit entspricht sie exakt der Wortlänge des XGMII Interface, welches zur Anbindung des MGT verwendet wird. Im Ruhezustand sendet der Transceiver eine Idlesignal. Die Transmission eines Pakets wird durch ein Startbyte eingeleitet, welches die Generierung einer Preamble zur Folge hat. Anschließend werden die aus  $n$  Datenwörtern bestehenden Nutzdaten des Paktes gesendet. Der Wert  $n$  ist ein Designparameter und daher zum Synthesezeitpunkt bereits festgelegt. Daher besitzen alle generierten Pakete dieselbe Größe, so dass sich eine konstante Latenz einstellt. Durch die Wahl eines kleineren Wertes für  $n$  kann eine kleinere Latenz erreicht werden, während ein höherer Wert in einem geringeren Protokolloverhead resultiert. Auf diese Weise kann ein optimales Verhältnis zwischen Latenz und Protokolloverhead ausgewählt

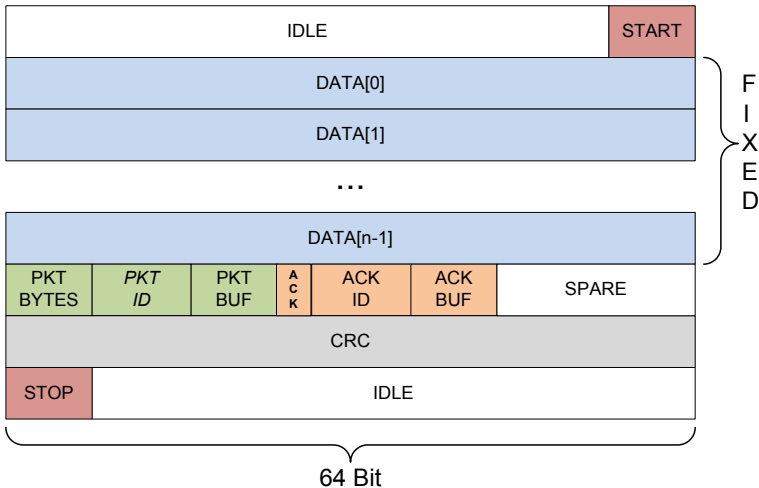


Abbildung 5.11.: Paketformat der Inter-Kommunikationsschnittstelle

werden. Den Nutzdaten folgt schließlich ein Kontrollwort und eine 64-Bit CRC Prüfsumme. Wenn keine weiteren Pakete gesendet werden müssen, wird ein Stoppbyte erzeugt und der MGT kehrt in den Ruhezustand zurück. Anderenfalls folgt das nächste Paket direkt nach dem CRC Wort.

Das Kontrollwort beinhaltet *lokale* Kontrolldaten (in grün eingezeichnet), welche Informationen zum aktuell gesendeten Paket beinhalten sowie *remote* Kontrolldaten (in orange eingezeichnet), die von korrekt empfangenen Paketen der entgegengesetzten Richtung herrühren. Das lokale Kontrolldatenfeld *PKT\_BYTES* beinhaltet die Anzahl der Nutzdatenwörter, die tatsächlich belegt sind. Da die Anzahl der Nutzdatenwörter in jedem Paket konstant ist, besteht die Möglichkeit, dass die Anzahl der belegten Wörter geringer als  $n$  ist, falls beim Erstellen des Pakets nicht genügend Nutzdaten bereitstanden um ein komplettes Paket zu füllen. Es besteht sogar die Möglichkeit, dass gar keine Nutzdaten übermittelt werden, aber korrekt empfangene Pakete der Gegenrichtung bestätigt werden müssen. Um die Latenz so gering wie möglich zu halten, wird ein Paket generiert, sobald Nutzdaten oder Bestätigungsinformationen zur Übertragung bereit stehen. Das lokale Kontrolldatenfeld *PKT\_ID* beinhal-

## 5. Realisierung der Signalverarbeitung

tet eine aufsteigende Paket ID, die es dem Empfänger erlaubt, fehlende Pakete innerhalb einer Sequenz zu erkennen. Das lokale Feld *BUF\_ID* enthält schließlich noch die Puffernummer des Senders, aus dem das Paket generiert wurde. Die Felder *PKT\_ID* und *BUF\_ID* werden zusammen mit weiteren Daten verwendet, um die Neuübertragung fehlerhafter oder verlorengegangener Pakete anzustoßen. Wenn ein Paket korrekt empfangen wurde werden diese beiden Werte daher im *remote* Bereich des Kontrollworts eines Pakets, welches in die entgegengesetzte Richtung übermittelt wird, an den Absender zurückgesendet.

### 5.6.3.3. Hardwarearchitektur

Ein Blockdiagramm, welches die realisierte Hardwarearchitektur der Inter-Kommunikationsschnittstelle zeigt, ist in Abbildung 5.12 zu sehen.

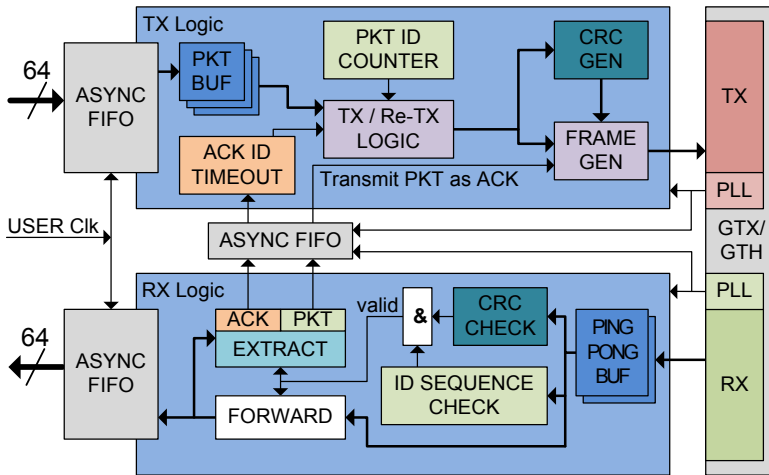


Abbildung 5.12.: Blockdiagramm der Inter-Kommunikationsschnittstelle

Das Nutzerdesign wird mit der Schnittstelle über zwei gewöhnliche asynchrone FIFOs verbunden. Diese sorgen für die Entkopplung der unterschiedlichen Taktdomänen des Nutzerdesigns und der Schnittstelle. So lange Datenwörter am FIFO Ausgang bereitstehen und mindestens ein

Sendepuffer frei ist, entnimmt die TX Logik maximal  $n$  Nutzdatenwörter aus dem FIFO und speichert sie im nächsten freien Sendepuffer. Die Anzahl der verfügbaren Sendepuffer, zur Designzeit konfigurierbar über den Parameter  $m$ , legt die maximale Anzahl an Paketen fest, die gesendet werden können, bevor eine Bestätigung empfangen werden muss.

Ein anderer Teil der *TX/Re-TX\_LOGIC* überprüft, ob ein Puffer bereit zum Versenden ist und fügt diesem eine aufsteigende Paket ID (*PKT\_ID*) und die Sendepuffernummer (*BUF\_ID*) hinzu, um ein Paket zu formen. Schließlich hängt der *FRAME\_GEN* Block die remote Kontrolldaten von der Empfangseinheit der Schnittstelle sowie eine CRC an das zu sendende Paket an und leitet es zum Sendeteil des MGT weiter. Sollte eine Zeitüberschreitung einer Paketbestätigung auftreten, da ein zuvor versendetes Paket nicht rechtzeitig bestätigt wurde, dann führt die *TX/Re-TX\_LOGIC* zunächst eine Neuübertragung der betroffenen Pakete durch.

In der Empfangseinheit der Schnittstelle passiert jedes vom MGT eintreffende Paket zunächst einen Ping-Pong Puffer. Wenn ein Paket vollständig empfangen wurde, wird es auf eine korrekte CRC Prüfsumme sowie auf eine aufsteigende und lückenlose Paket ID überprüft. Wenn beide Tests erfolgreich verlaufen und das ausgehende asynchrone FIFO genügend freien Speicherplatz besitzt, dann werden die Nutzdaten des empfangenen Pakets zum FIFO weitergeleitet, ansonsten wird das Paket verworfen. Währenddessen erlaubt der zweite Teil des Ping-Pong Puffers den unterbrechungsfreien Empfang eines eventuell direkt nachfolgenden Pakets.

Gleichzeitig werden die Paket und Puffer ID aus den *lokalen* Kontrolldaten des zuletzt empfangenen Pakets zur TX Logik weitergeleitet, wo sie dem *remote* Kontrolldaten Bereich des nächsten ausgehenden Pakets hinzugefügt werden. Ähnlich werden auch die Paket und Puffer ID aus den *remote* Kontrolldaten des zuletzt empfangenen Pakets zur Sendewiederholungslogik (*TX/Re-TX\_LOGIC*) der Sendeeinheit weitergeleitet. Hier werden diese Daten genutzt, um zu überprüfen, ob ein Sendepuffer freigegeben werden kann und für neue Daten bereitsteht. Der Kontrolldatentransfer der Empfangseinheit zur Sendeeinheit nutzt ein weiteres asynchrones FIFO, um den Übergang zwischen den verschiedenen Taktdomänen der beiden Bereiche zu ermöglichen.





## 6. Bereitstellung der Hardwareplattform

Wie bereits in Abschnitt 4.1.3.1 von Kapitel 4 erläutert wurde, existiert leider keine kommerzielle Hardwareplattform, welche sämtliche erforderlichen Anforderungen erfüllt. Dabei ist insbesondere die Menge der verfügbaren GTX und GTH Transceiver nicht ausreichend und der Aufbau einer geeigneten Entwicklungs- und Testplattform ist für die Demonstration des Kommunikationssystems daher zwingend erforderlich. Zusätzlich ermöglicht sie die Integration spezifischer Schnittstellen für dieses Projekt direkt auf dem FPGA Mainboard. Dazu gehören z.B. die Sockel für die eingesetzten 25 GSa/s ADCs und DACs sowie die SFP+ Einschübe für die 10G Ethernet Transceiver. Verglichen mit einer Lösung, die auf Standardhardware setzt und somit viele verschiedene Platinen kombiniert, ergeben sich so eine Reihe von Vorteilen: Weniger Verkabelungsaufwand, kürzere Leitungen mit besserer Impedanzanpassung und weniger Steckverbinder verbessern die Signalqualität deutlich. Außerdem existiert weniger Spielraum für fehlerhafte Verbindungen und Wackelkontakte und es ergibt sich ein insgesamt kompakteres und transportableres System.

Nichts desto trotz beinhaltet das hier vorgestellte FPGA Board aus verschiedensten Gründen nicht alle für den Betrieb des Kommunikationssystems benötigten Hardwarekomponenten. Einige Teile wie z.B. die ADCs sowie DACs der ONUs oder auch die DDS Taktquelle für deren Betrieb wurden aus Flexibilitäts- und Wiederverwertbarkeitsgründen in Form von externen Modulen realisiert. Diese Zusatzmodule werden genauso wie das Mainboard im Folgenden in diesem Kapitel betrachtet.

### 6.1. FPGA Mainboard

Das eigentliche Design des FPGA Mainboards sowie dessen Produktion und Bestückung wurde extern durchgeführt. Die Spezifikation und Inbetriebnahme sind jedoch Teil dieser Arbeit, so dass hier eine kurze Übersicht über diese beiden Teile gegeben werden soll.

Aufgrund von Überlegungen, die in Abschnitt 4.1.3.1 diskutiert wurden, bildet ein Virtex-6 FPGA vom Typ XC6VHX565T in der 1924 Pin-Version das zentrale Element des Mainboards. Alternativ kann auch der Typ XC6VHX380T eingesetzt werden, da er in einem identischen Gehäuse und pinkompatibel verfügbar ist. Wie bereits in Kapitel 4 erläutert wurde, müssen mindestens drei 64-Bit SO-DIMM DDR3-1066 Speichermodule für den benötigten Datendurchsatz angebunden werden. Da die Speichermodule über ein schnelles DDR Interface verfügen, sind nur bestimmte Pinanordnungen am FPGA nutzbar. Für genaue Details dazu sei auf den Xilinx Memory Interface Generator (MIG)-Userguide [45] verwiesen. Aufgrund dieser Einschränkungen ist die Speicheranbindung verhältnismäßig kompliziert und wurde daher als Erstes bei der Entwicklung des Schaltplans definiert.

Unter Zuhilfenahme des Xilinx MIG wurde die endgültige Speicherpinbelegung am FPGA zusammen mit dem externen Dienstleister in vielen Iterationen ermittelt. Dabei mussten einige Designparameter parallel beachtet und optimiert werden, so dass sich sehr viele zu berücksichtigende Einschränkungen ergaben: Jedes 64 Bit breite Speicherinterface benötigt neben den eigentlichen Signalpins für den Daten-, Adress- und Steuerbus noch zusätzliche freie Pins am FPGA für interne Abläufe. Je nach Aufteilung des Interfaces auf verschiedene IO Bänke des FPGAs werden unterschiedliche Mengen dieser Zusatzpins benötigt. Weiterhin wird die IO Spannung der beteiligten Bänke auf 1.5V festgelegt, so dass für die Anbindung zusätzlicher Peripheriehardware eventuell Pegelwandler benötigt werden. Somit war hier eine Aufteilung der Speicherinterfaces auf möglichst wenige Bänke wünschenswert. Zusätzliche zu berücksichtigende Faktoren sind z.B. eine gute Routebarkeit der Signale auf der Platine, eine maximale Anzahl parallel schaltender IO Pins pro Bank oder die verschiedenen Anforderungen, welche auch von anderen Peripheriekomponenten an die FPGA Anbindung gestellt werden.

Neben der Speicheranbindung ist auch die Verbindung der ADCs, DACs und 10G Ethernet Transceiver mit dem FPGA kritisch. Hier kommen jedoch dedizierte IO Pins der GTX und GTH Transceiver zum Einsatz, so dass sich zumindest bei der Schaltplanentwicklung nur relativ wenige Wahlmöglichkeiten ergeben. Die Komplexität liegt vor allem beim Layout der MGL Signalleitungen, die im Fall der GTH Transceiver bei einer Bandbreite von mehr als 5 GHz arbeiten müssen. Es sollte an dieser Stelle erwähnt werden, dass beim Design des Mainboards lediglich die Anbindung der 25 GSa/s ADCs und DACs von Micram berücksichtigt wurde. Die langsameren 3.125 GSa/s Wandler für den Betrieb des Mainboards als ONU werden über Adapterplatinen in dieselben Sockel eingesteckt und müssen entsprechend (hardware-)kompatibel designt werden. Details dazu folgen weiter unten in Abschnitt 6.3 dieses Kapitels.

Die Taktversorgung der GTX und GTH Transceiver ist sehr flexibel konfigurierbar. Mehrere kaskadierte Multiplexer erlauben die individuelle Auswahl verschiedener On- sowie Offboard Taktquellen. Dazu gehören z.B. der Referenztakt der ADCs und DACs, ein über Sub-Miniature-A (SMA) Anschlüsse zugeführter externer Takt oder ein lokaler Quarzoszillator. So ergibt sich eine große Auswahl verschiedener Taktquellen und damit eine hohe Flexibilität, so dass beim späteren Betrieb idealerweise keinerlei Engpässe an dieser Stelle auftreten.

Neben diesen sehr speziellen und auf das Kommunikationssystem ausgelegten Schnittstellen besitzt das Mainboard noch einige weitere typische Komponenten. Dazu gehören mehrere Taktgeneratoren, ein Konfigurationsspeicher, drei serielle Schnittstellen, ein 1G Ethernet Port und ein SD Karten Slot sowie eine FMC Schnittstelle zur Anbindung kommerzieller Erweiterungskarten oder Eigenentwicklungen. Weiterhin sind noch ein Complex Programmable Logic Device (CPLD) sowie ein Mikrocontroller vorhanden. Der CPLD erlaubt das flexible Verbinden der Signalleitungen von Taktgeneratoren, Taktmultiplexern, Konfigurationsspeicher und vielen weiteren Hilfsbausteinen wie z.B. kleinen DACs zur Erzeugung von Bias-Spannungen für die schnellen Micram ADCs und DACs. Der Mikrocontroller erlaubt darüber hinaus die eigenständige und vom FPGA unabhängige Übernahme von einfachen Aufgaben wie z.B. der Lüftersteuerung für die Wandler und den FPGA.

## 6. Bereitstellung der Hardwareplattform

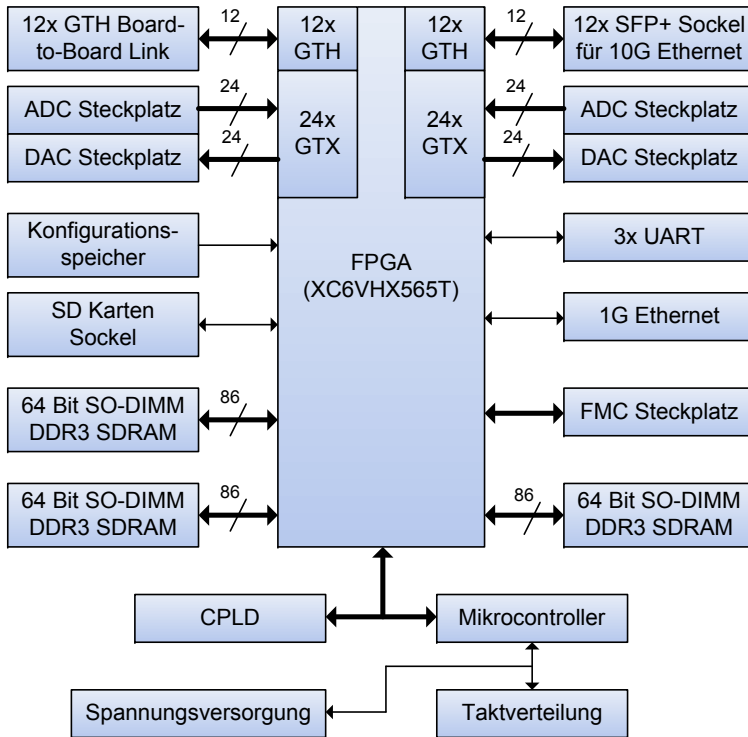


Abbildung 6.1.: Blockdiagramm des FPGA Mainboards

Zu guter Letzt verfügt das Mainboard noch über drei Hochgeschwindigkeitsschnittstellen zur direkten Vernetzung mehrerer FPGAs untereinander. Jede dieser Schnittstellen führt vier voll duplex fähige GTH Links. Insgesamt stehen einem Mainboard somit zwölf mal 10 GBit/s bidirektional für die Board-to-Board Kommunikation zur Verfügung. Das ist wichtig, da selbst der ressourcenreiche FPGA des Mainboards nicht gleichzeitig zum Betrieb des Senders und Empfängers am OLT ausreicht und daher zum Betrieb der Vermittlungsstelle mehrere Board kombiniert werden müssen. Abbildung 6.1 zeigt ein Blockdiagramm des entwickelten FPGA Mainboards.

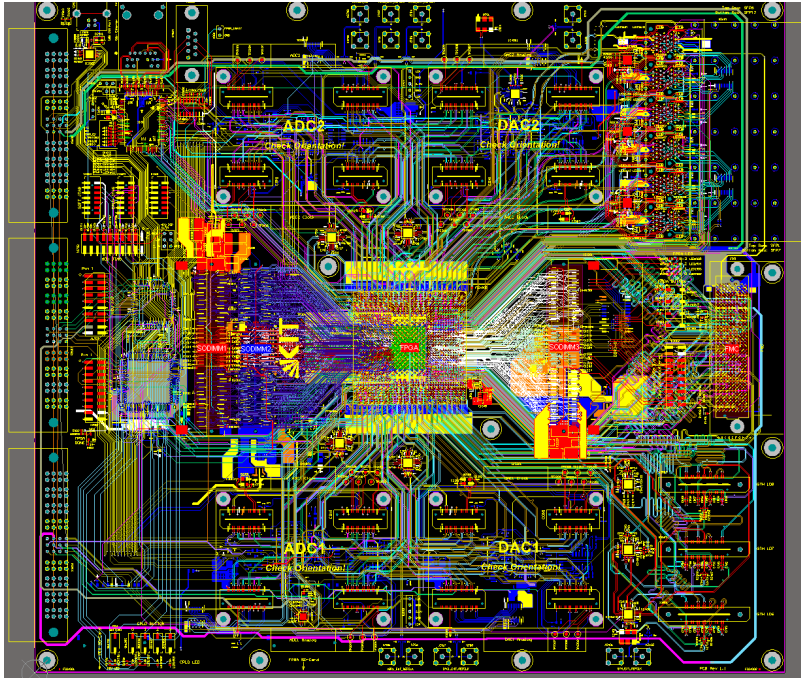


Abbildung 6.2.: Layout des FPGA Mainboards

Nach der gemeinsamen Entwicklung des Schaltplans erfolgte die Erstellung des Layouts weitgehend selbstständig durch den externen Dienstleister. Die Hauptschwierigkeit dabei war die Erstellung eines tragfähigen Konzepts zur Stromversorgung des FPGA und der Wandler sowie das impedanz- und längenkontrollierte Routing der GTX-, GTH- und Speichersignalleitungen. Das resultierende Platinenlayout verfügt über 24 Lagen. Allein daran ist die hohe Komplexität der Entwicklung ablesbar. Abbildung 6.2 zeigt einen Screenshot des fertigen Layouts. Zur besseren Übersicht sind alle Spannungsversorgungslagen ausgeblendet.

Um das Produktionsrisiko zu minimieren wurden zunächst nur zwei Platinen gefertigt und systematisch getestet. Es zeigte sich sehr schnell, dass die wesentlichen Komponenten des Boards wie erwartet funktionieren.

## 6. Bereitstellung der Hardwareplattform

---

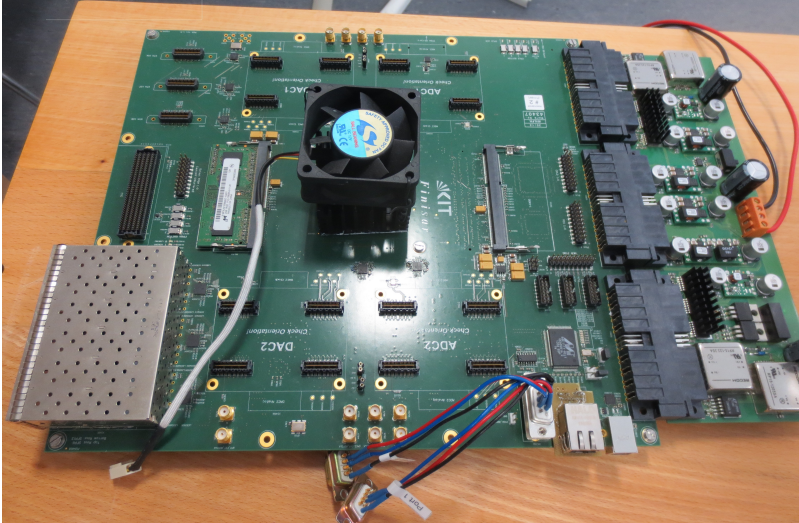


Abbildung 6.3.: FPGA Mainboard

Kleinere Probleme durch fehlerhaft definierte Verbindungen bzw. durch falsch beschaltete Terminierungsnetzwerke von IO Anschlüssen wurden beim 1G Ethernet Port und den Taktmultiplexern ermittelt. Alle Fehler konnten manuell durch nachträgliche Modifikation der Verbindungen korrigiert werden, so dass ein vollständig einsatzbereites Mainboard als Referenz und für weitere Tests des Kommunikationssystems zur Verfügung stand. Nachdem die ermittelten Fehler in Schaltplan und Layout korrigiert wurden, konnten auch die restlichen fünf Boards zur Produktion freigegeben werden. Abbildung 6.3 zeigt ein fertig aufgebautes Mainboard auf dem deutlich sichtbar die vier Steckplätze für die beiden AD sowie DA Wandler zu erkennen sind.

### 6.2. Stromversorgung

Das FPGA Mainboard wurde bewusst ohne Module zur Stromversorgungen designt. Bis auf wenige, für die Erzeugung von sehr genauen

Referenzspannungen benötigte Spannungsregler, befinden sich sämtliche Komponenten der Stromversorgung auf einer zweiten, externen Platine. Die gewählte Aufteilung hat den Vorteil, dass z.B. die Spannungsregler für die Versorgung der empfindlichen DA und AD Wandler bei Bedarf einfach ausgetauscht werden können, falls die erreichte Versorgungsspannungsstabilität nicht ausreicht. Weiterhin ist die Produktion des 24 lagigen Mainboards sehr kostenintensiv und die Auslagerung der Energieversorgung auf eine zweite Platine spart eine Menge Platinenfläche und somit Kosten ein. Im Gegensatz zum Mainboard wurde das Stromversorgungsmodul komplett als Teil dieser Arbeit erstellt.

Aufgrund der Vielfalt an komplexen Komponenten die sich auf dem Mainboard befinden, ergeben sich entsprechend hohe Anforderungen an die Anzahl und Leistungsfähigkeit der bereitzustellenden Versorgungsspannungslevel. Tabelle 6.1 fasst die Anforderungen an die Stromversorgung des FPGA Mainboards zusammen.

Die angegebenen Maximalströme ergeben sich aus den Datenblättern der eingesetzten Komponenten. Es kann sicherlich davon ausgegangen werden, dass nicht alle Versorgungszweige permanent und simultan mit dem Maximalstrom belastet werden, da es sehr unwahrscheinlich ist, dass alle Komponenten gleichzeitig auf höchstem Leistungslevel arbeiten. Trotzdem zeigt Tabelle 6.1 eindrucksvoll, welche Leistungsaufnahme ein komplexes und hochperformantes Signalverarbeitungssystem besitzt. Interessant ist an dieser Stelle, dass der größte Anteil der Leistungsaufnahme nicht unbedingt von der eigentlichen digitalen Logik ausgeht (FPGA Core: maximal 40 W), sondern die Hochgeschwindigkeitswandler ADC/DAC mit maximal 116.8 W mehr als doppelt soviel Leistung benötigen. Auch die Kommunikation mit Hilfe der MGLs ist mit etwa 19 W nicht gerade sparsam.

Zur Erzeugung der vielen verschiedenen Spannungslevel kommen unterschiedliche Verfahren zum Einsatz. Unempfindliche und stromhungrige Zweige wie beispielsweise *FPGA Core*, *FPGA Auxiliary* oder die verschiedenen *General & Digital-IO* Level werden direkt mit effizienten Schaltreglern versorgt. Dazu wurden fertige Module des Typs PTH08T210WAD, PTH08T220WAD sowie PTH08T230WAD von Texas Instruments eingesetzt. Diese leisten einen Ausgangsstrom von bis zu 30A, 16A bzw. 6A bei variabler Ausgangsspannung. Weiterhin besitzen die Module einen

## 6. Bereitstellung der Hardwareplattform

---

Kontrolleingang zum definierten Hoch- und Herunterfahren der Ausgangsspannung beim Ein- und Ausschalten des Systems. So kann garantiert werden, dass die von den Verbrauchern geforderten Power-Up bzw. Power-Down Sequenzen eingehalten werden. In empfindlicheren Zweigen, die zur Versorgung von analogen Schaltungskomponenten genutzt werden (dazu gehört auch die Versorgung der MGT Schnittstellen), kommen hingegen Linearregler zum Einsatz. Im Fall von geringer benötigter Leistung wie z.B. bei den  $\pm 5V$  der *Analogue support* Schaltungen kann die Versorgung ausschließlich über Linearregler direkt aus der 12V Eingangsspannung der Stromversorgungsplatine gewonnen werden. Die anfallende Verlustleistung hält sich trotz einer Spannungsdifferenz von 7V in Grenzen. Problematischer ist die Bereitstellung der Versorgung für die Multi-Gigabit Transceiver. Hier kommt zunächst ein Schaltregler zum Einsatz, der eine Zwischenspannung von 2.2V mit hoher Effizienz bereitstellt. Anschließend sorgen verschiedene Linearregler für die weitere Reduktion der Spannung auf die benötigten Werte der einzelnen Versorgungsarme. Hierzu kommen ICs des Typs MIC69502 von Micrel zum Einsatz. Sie leisten einen Ausgangsstrom von maximal 5A bei ebenfalls einstellbarer Ausgangsspannung. Für die Erzeugung der negativen Spannungen wurden galvanisch isolierende DC-DC Wandler des Typs RP15-123.3SA bzw. RP15-1205SA mit einem nachgeschalteten Linearregler vom Typ LT1185 eingesetzt.

Der hohe Anteil fertiger Module beim Design der Stromversorgung erfolgte einerseits zur Reduktion der Komplexität des Aufbaus. Andererseits wird so auch eine hohe Qualität der Ausgangsspannungen garantiert und die Fehlerwahrscheinlichkeit beim Aufbau des Stromversorgungsmoduls sinkt deutlich. In Anbetracht der hohen Kosten der Komponenten des FPGA Mainboards ist dies ein besonders wichtiger Faktor.

Beim Layout der Platine musste vor allem auf ausreichende Stromtragfähigkeit der Leiterbahnen geachtet werden. Unterstützend dazu wurde eine 70  $\mu m$  starke Kupferauflage der Platine gewählt. Das Layout kommt mit zwei Lagen aus, wobei die untere Lage primär als Massefläche dient und nur vereinzelt durch Brücken für Leiterbahnen der oberen Lage unterbrochen wird. Abbildung 6.4 zeigt ein fertig bestücktes Stromversorgungsmodul für ein FPGA Mainboard.



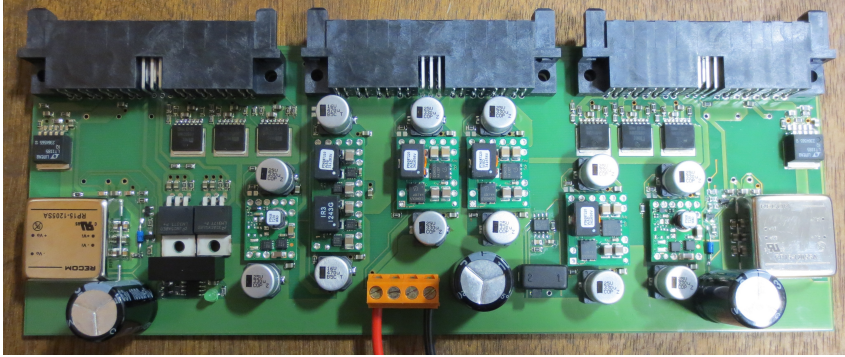


Abbildung 6.4.: Mainboard Spannungsversorgung

### 6.3. DA und AD Wandler für die Teilnehmeranschlüsse

Wie bereits in Abschnitt 6.1 dieses Kapitels erläutert, lassen sich lediglich die 25 GSa/s AD und DA Wandler direkt mit dem FPGA Mainboard verwenden. Für den Einsatz der 3.125 GSa/s Wandler werden noch entsprechende Adapterplatinen benötigt, welche ebenfalls als Teil dieser Arbeit entwickelt, produziert und getestet wurden.

In einem ersten Schritt erfolgte zunächst die Auswahl geeigneter ADCs und DACs. Aus der Verfügbarkeit der DACs und ADCs des OLT die bei einer Abtastfrequenz von 25 GSa/s arbeiten könnte man schließen, dass Wandler im einstelligen GSa/s Bereich weit verbreitet und leicht verfügbar sind. Tatsächlich bilden die Micram Wandler aber eine Ausnahme, die speziell für die Forschung im Bereich der optischen Kommunikationstechnologien entwickelt wurde. Diese Wandler werden auch nicht in großen Stückzahlen hergestellt sondern besitzen mehr den Status eines Prototypen oder einer Kleinserie zur Ermittlung eines eventuell zukünftigen Marktpotentials. Tatsächlich war die Auswahl geeigneter DA und AD Wandler im benötigten Bereich von 3.125 GSa/s zum Designzeitpunkt dieser Arbeit stark eingeschränkt. Somit war einfach aufgrund der Ver-

## 6. Bereitstellung der Hardwareplattform

---

fügbare Produkte recht schnell klar, welche Wandler für die Verwendung an den ONUs zum Einsatz kommen.

Für den DAC fiel die Wahl auf den Baustein MAX5881 von Maxim-Integrated. Als ADC kommt hingegen das IC ADC12D1800 von National Semiconductor (mittlerweile von Texas Instruments aufgekauft) zum Einsatz. Beide Bausteine besitzen eine Auflösung von 12 Bit und verfügen über einen differentiellen Sampletakteingang sowie einen differentiellen Analogein- bzw. ausgang. Die Ein- bzw. Ausgabe der digitalen Datenströme erfolgt bei beiden Wandlertypen über vier gemultiplexte, 12 Bit breite Datenbusse im DDR Modus bei differentieller LVDS Signalisierung. Weiterhin gibt jeder Wandler als zeitlichen Bezugspunkt für den digitalen Datenstrom noch ein differentielles Taktsignal aus, welches vom FPGA zum Betrieb der DDR Schnittstelle verwendet wird. Entsprechend hoch ist die Anzahl der vorhandenen Pins für die digitale Schnittstelle. Insgesamt 48 LVDS Paare für Daten und eins für den Takt, also 98 Pins, müssen im DDR Modus versorgt werden.

Dem gegenüber stehen die 24 differentiellen Leitungspaare der MGTs, die an jedem ADC bzw. DAC Sockel des FPGA Mainboards bereitstehen. Somit stehen also lediglich die Hälfte der benötigten Leitungspaare pro Sockel zur Verfügung. Nach der Evaluierung verschiedener Möglichkeiten wie z.B. der Verwendung von 2:1 Multiplexern wurde vor allem aus Gründen einer erhöhten Flexibilität entschieden, jeweils einen FPGA pro Adapterplatine als Bindeglied zwischen den MGLs der Sockel und den eigentlichen Wandlern einzusetzen. Dieses FPGA kann dann auch gleichzeitig für zusätzliche Aufgaben wie z.B. zur zeitlichen Verschiebung des einzulesenden bzw. auszugebenden Datenstroms eingesetzt werden, wodurch Ressourcen im FPGA des Mainboards eingespart werden können. Für das einzusetzende FPGA fiel die Wahl auf einen 6VLX75TFF484, ebenfalls ein Typ der Xilinx Virtex-6 Serie. Seine Auswahl basiert vor allem auf der verfügbaren Datenrate die dieser FPGA durch seine GTX Transceiver, die entsprechend direkt mit den GTX Transceivern des Mainboard FPGA verbunden werden können, bereitstellt. Bei einer Auflösung von 12 Bit und einer Samplerate von 3.125 GSa/s ergibt sich ein Durchsatz von 37.5 GBit/s. Entsprechend werden mindestens sechs Transceiver bei einer Datenrate von 6.5 GBit/s benötigt, wobei es sich bei dem gewählten FPGA um den kleinsten verfügbaren Typ der Virtex-6 Serie handelt, welcher diese Anforderungen erfüllt. Um Kosten einzusparen, wurde parallel auch

### 6.3. DA und AD Wandler für die Teilnehmeranschlüsse

der Einsatz eines Virtex-5 Typs evaluiert, dieser besitzt aber lediglich GTP Transceiver, die nur bis zu 3.25 GBit/s spezifiziert sind. Die Kosten eines FPGA aus der Virtex-5 Serie mit einer entsprechend hohen Anzahl an GTP Transceivern hätten daher über denen des Virtex-6 Modells gelegen. Der gewählte FPGA wird zusammen mit dem entsprechenden DA bzw. AD Wandler auf einer kleinen Adapterplatine platziert, die über entsprechende Steckverbinder direkt auf die Sockel des Mainboards aufgesteckt werden kann. Die digitale Schnittstelle des Wandlers wird direkt mit normalen IO Pins des FPGA verbunden. Neben den beiden Bausteinen werden noch einige Schalt- und Linearregler zur Spannungsversorgung benötigt, die ebenfalls auf der Platine Platz finden müssen. Abbildungen 6.5 und 6.6 zeigen das jeweilige Blockdiagramm eines ADC bzw. DAC Moduls.

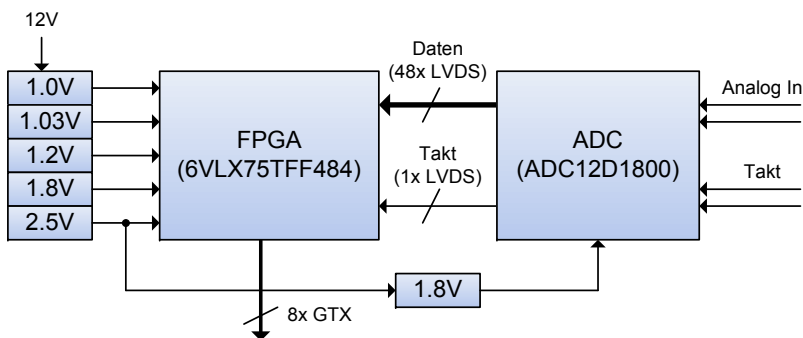


Abbildung 6.5.: Blockdiagramm eines ADC Moduls einer ONU

Zum Betrieb des FPGA wird dabei eine Spannung von 1.0V für den Kern benötigt. Die Versorgung der GTX Transceiver erfordert 1.03V und 1.2V. Weiterhin werden noch 2.5V für die Auxiliaryspannung und 1.8V als IO-Spannung benötigt. Sämtliche Spannungen werden mit Hilfe von effizienten Schaltreglern aus einer 12V Eingangsspannung des Moduls generiert.

Zum Betrieb des ADC ADC12D1800 werden ebenfalls 1.8V erzeugt. Zur Erhöhung der Qualität der Versorgungsspannung erfolgt dies jedoch mit Hilfe eines Linearreglers aus der 2.5V Versorgung der FPGA-

## 6. Bereitstellung der Hardwareplattform

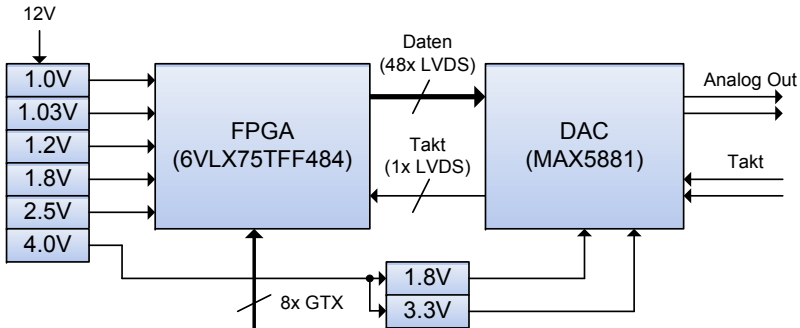


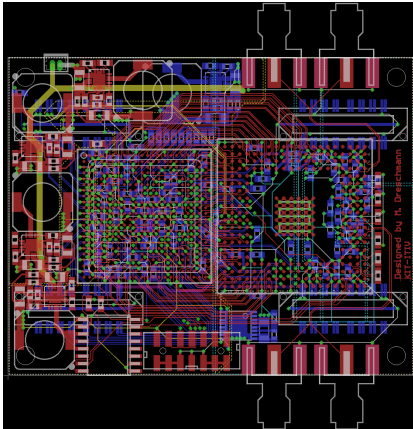
Abbildung 6.6.: Blockdiagramm eines DAC Moduls einer ONU

Auxiliaryspannung. Auch die Versorgung des DAC MAX5881 erfolgt aus Linearreglern. Da dieser jedoch 1.8V und 3.3V benötigt, wird auf DAC Modulen zusätzlich noch eine Zwischenspannung von 4V mit Hilfe eines weiteren Schaltreglers generiert.

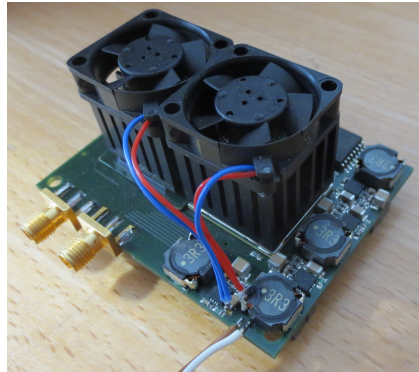
Sowohl Auswahl der Bauteile, die Entwicklung der Schaltpläne sowie Layout, Bestückung und Test der Module wurde vollständig als Teil dieser Arbeit durchgeführt. Abbildung 6.7 zeigt die Layouts der Module sowie zwei fertig bestückte Platinen.

### 6.4. DDS Taktgenerator

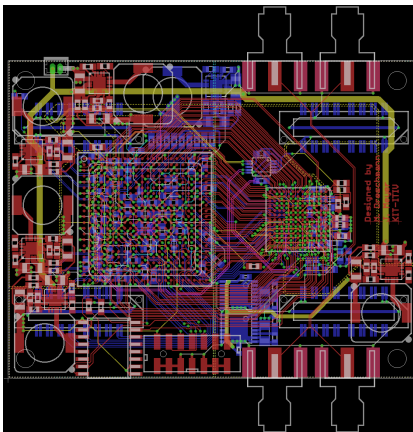
In Abschnitt 4.3.1 von Kapitel 4 wurde unter anderem die Synchronisation des Sampletakts der ONUs erläutert. Dazu muss die Abtastrate der Wandler sehr fein einstellbar sein. Im Gegensatz zur Kompensation des Lokaloszillatoroffsets ist eine feine Modifikation der Abtastrate mit Hilfe von Resampling Algorithmen nur mit hohem Aufwand und ressourcenintensiv in der digitalen Domäne möglich. Aus diesem Grund wurde für das hier vorgestellte Kommunikationssystem ein anderer Weg gewählt. Anstatt eines digitalen Resamplings wird direkt die Frequenz des Taktgenerators für die AD und DA Wandler der ONU modifiziert. Da die Wandler des ONU sowohl bei der steigenden als auch bei der



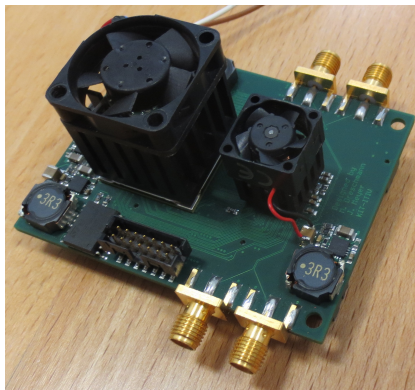
(a) Layout eines ADC Moduls



(b) ADC Modul einer ONU



(c) Layout eines DAC Moduls



(d) DAC Modul einer ONU

Abbildung 6.7.: ADC und DAC einer ONU

## 6. Bereitstellung der Hardwareplattform

---

fallenden Taktflanke ein Sample aufnehmen, wird für eine Samplerate von 3.125 GSa/s ein Takt von 1.5625 GHz benötigt.

Ein solcher, variabler Taktgenerator wird heute typischerweise in Form einer DDS aufgebaut. Allerdings war zum Designzeitpunkt dieses Oszillators kein integrierter DDS Baustein mit ausreichend hoher Ausgabefrequenz verfügbar, so dass er direkt als Taktquelle für die Wandler hätte dienen können. Zur Lösung dieses Problem bot sich daher an, zunächst mit Hilfe einer DDS eine variable Zwischenfrequenz zu generieren und diese anschließend mit einer nachfolgenden PLL auf die benötigte Zielfrequenz hochzusetzen.

Als DDS Baustein kommt ein AD9910 von Analog Devices vom Einsatz. Dabei handelt es sich um einen Baustein mit einem 14 Bit DA Wandler, der bei 1 GSa/s arbeitet und so Frequenzen bis zu 400 MHz erzeugt. Die für den DAC benötigten 1 GHz werden über eine im IC integrierte PLL aus einem 25 MHz Quarzoszillator gewonnen. Das Phasenakkumulatorregister besitzt eine Breite von 32 Bit, woraus sich bei einer Abtastfrequenz von 1 GSa/s eine Frequenzauflösung von 0.23 Hz ergibt. Das IC verfügt noch über viele weitere Funktionen, z.B. zur Modulation von Frequenz, Phase und Amplitude. Diese werden jedoch in dieser Anwendung nicht benötigt.

Als Ausgabefrequenz der DDS wurde 125 MHz  $\pm$  3.90625 MHz gewählt. Das Ausgangssignal des DDS Bausteins passiert einen Antialiasing Tiefpass Filter aus drei kaskadierten LC-Gliedern und erreicht anschließend die PLL. Dazu kommt der Baustein AD9520 zum Einsatz, ebenfalls von Analog Devices. Es handelt sich dabei um eine PLL mit nachfolgender Taktverteilung. So können alle vier an einer ONU involvierten Wandler simultan mit demselben Takt versorgt werden.

Die Kommunikation sowohl des DDS als auch des PLL Bausteins mit den FPGA des Mainboards erfolgt über eine Serial Peripheral Interface (SPI) Schnittstelle. Weiterhin ist das DDS IC mittels eines 16 Bit breiten, parallelen Bus angebunden. Dieser wird genutzt, um die zu generierende Sollfrequenz (125 MHz) mit möglichst geringer Latenz zu modifizieren. Abbildung 6.8 zeigt das Layout der entwickelten Platine des Taktgenerators, auf Abbildung 6.9 ist die fertig bestückte Platine zu erkennen.

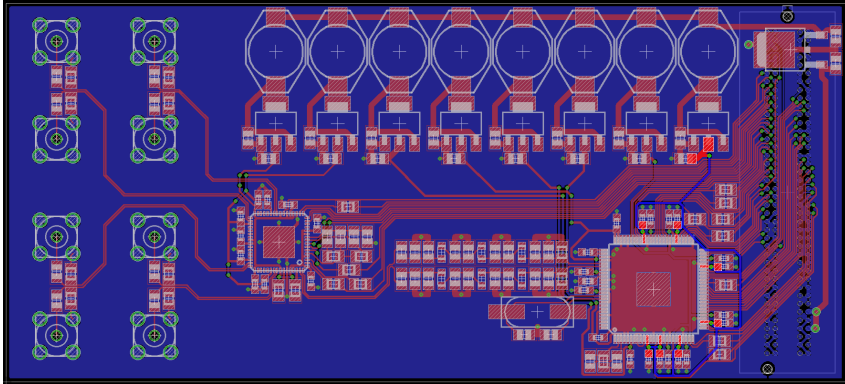


Abbildung 6.8.: Layout des DDS Taktgenerators

## 6.5. Analog Frontends

Bisher wurden lediglich Hardwarekomponenten des digitalen Signalverarbeitungssystems vorgestellt. Zum Abschluss dieses Kapitels soll jedoch noch kurz auf einige analoge Komponenten des Kommunikationssystems eingegangen werden. Abbildung 6.11 zeigt ein analoges Frontendboard, wie es bei einer ONU zum Einsatz kommt. Sein funktionaler Aufbau ist in Abbildung 6.10 in Form eines Blockdiagramms dargestellt. Zur

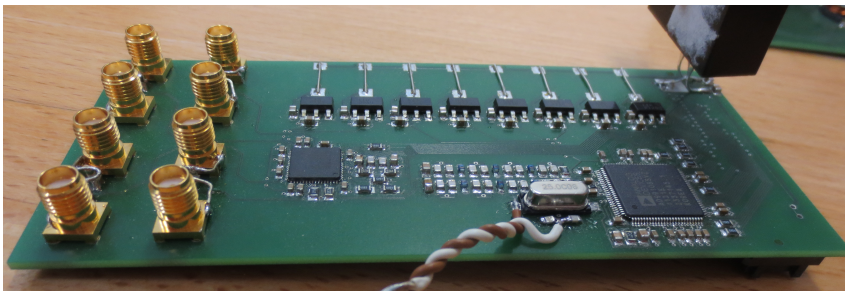


Abbildung 6.9.: Fertig aufgebauter DDS Taktgenerator

## 6. Bereitstellung der Hardwareplattform

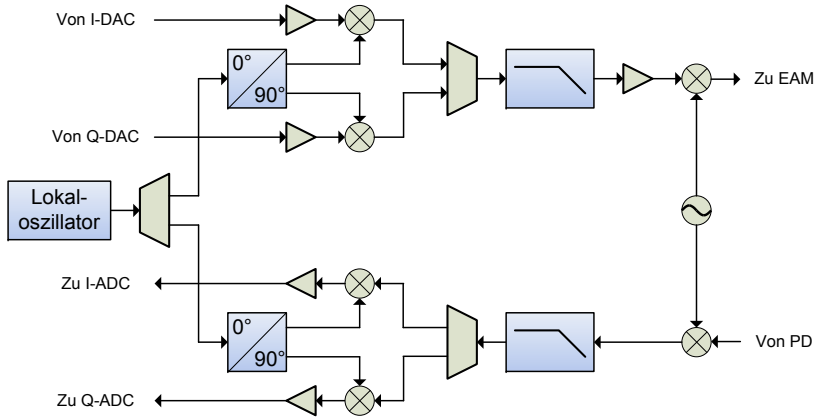


Abbildung 6.10.: Blockdiagramm der analogen Mischstufen der ONUs

Erhöhung der Wiederverwertbarkeit in nachfolgenden Projekten wurde das gesamte Board aus diskreten Komponenten zusammengesetzt.

Das in einem gemeinsamen Oszillator erzeugte Lokaloszillatorsignal (in Abbildung 6.11 hinten rechts zu erkennen) wird zunächst aufgeteilt und dem Downlink- bzw. Uplinkpfad zur Verfügung gestellt. Zum Betrieb der IQ Mischer wird nun zusätzlich ein um 90 Grad phasenverschobenes Oszillatorsignal in jedem Pfad generiert. Das Hoch- bzw. Herabmischen des ONU Up- bzw. Downlink Signals erfolgt schließlich mittels zweiter diskreter Mischer und den beiden Oszillatorsignalen. Weitere zu erkennende Komponenten sind Filter, Dämpfungsglieder und Verstärker, welche die auftretenden Signalleistungen an jedem Punkt der Schaltung in einem optimalen Arbeitsbereich halten und unerwünschte Mischprodukte entfernen.

Ein solches, in Abbildung 6.11 dargestelltes Frontendboard bildet das Bindeglied zwischen den ADCs bzw. DACs einer ONU sowie der Photodiode bzw. dem EAM welche die ONU an das Glasfasernetz koppeln.

Abbildung 6.12 zeigt schließlich noch einen der analogen Antialiasing Filter, welche am ONU an den AD und DA Wandlern zum Einsatz kommen. Bemerkenswert ist, dass diese ohne konventionelle Bauteile auskommen



und die Filterwirkung ausschließlich auf einer geschickten Anordnung der Leiterbahnen beruht.

## 6. Bereitstellung der Hardwareplattform

---

Komponente	Spannung	Maximalstrom
<b>ADC/DAC &amp; Analoge Hilfsschaltungen</b>		
ADC/DAC analogue front-end	-4.50 V	12.0 A
ADC/DAC analogue front-end	-3.30 V	16.0 A
Analogue support (OPV, etc)	-5.00 V	1.0 A
Analogue support (OPV, etc)	+5.00 V	1.0 A
<b>Multi-Gigabit Transceiver</b>		
GTH PLL	+1.80 V	2.00 A
GTH Receiver	+1.10 V	2.00 A
GTH Termination	+1.20 V	1.00 A
GTH Logic	+1.10 V	5.00 A
GTX Termination	+1.20 V	3.00 A
GTX Logic	+1.00 V	3.00 A
<b>FPGA</b>		
FPGA Core	+1.00 V	40.0 A
FPGA Auxiliary	+2.50V	10.0 A
<b>General &amp; Digital-IO</b>		
Lowspeed DAC/ADC Supply & FMC	+12.00 V	5.0 A
DAC/ADC Fan	+5.00 V	1.5 A
Digital IO, FMC	+2.50 V	5.0 A
Digital IO, FMC	+3.30 V	5.0 A
Digital IO, SD-Card	+5.00 V	1.0 A
Digital IO, DDR3, ADC/DAC	+1.50V	12.0 A

Tabelle 6.1.: Mainboard Spannungsversorgung

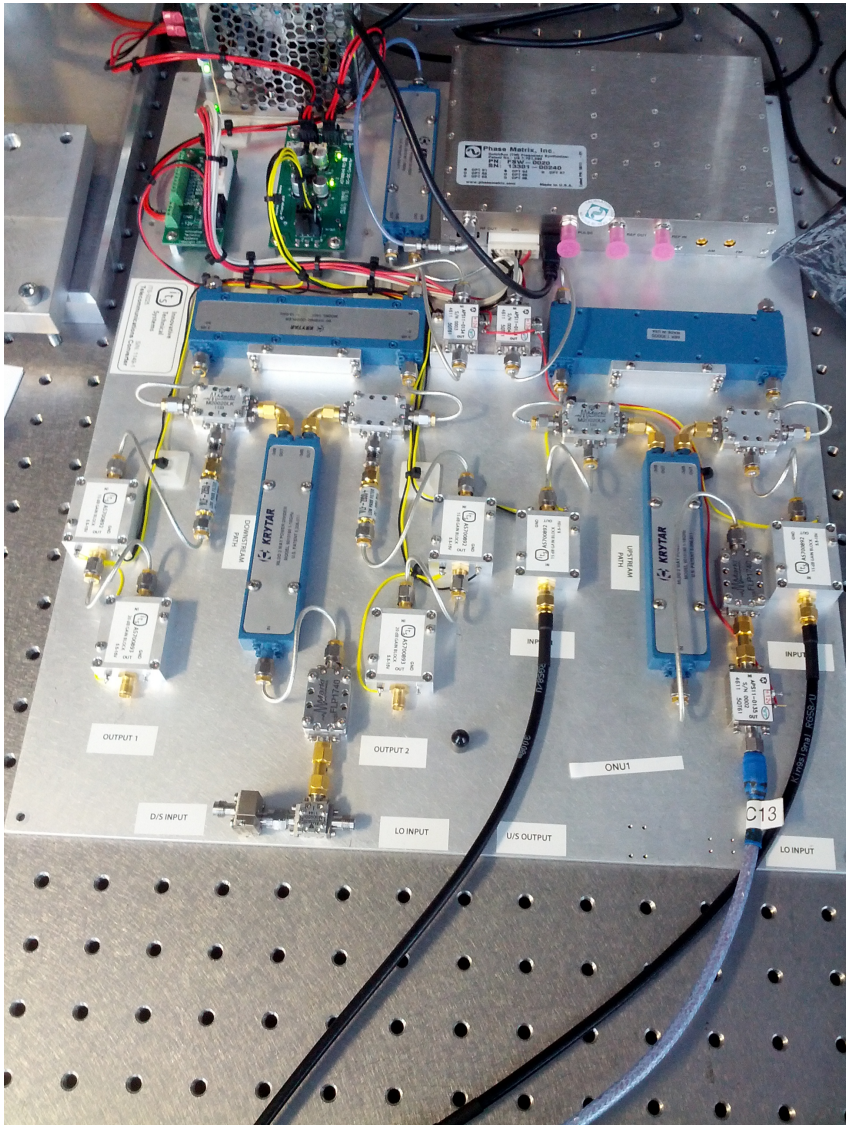


Abbildung 6.11.: Analoge Mischstufen der ONUs

## 6. Bereitstellung der Hardwareplattform

---

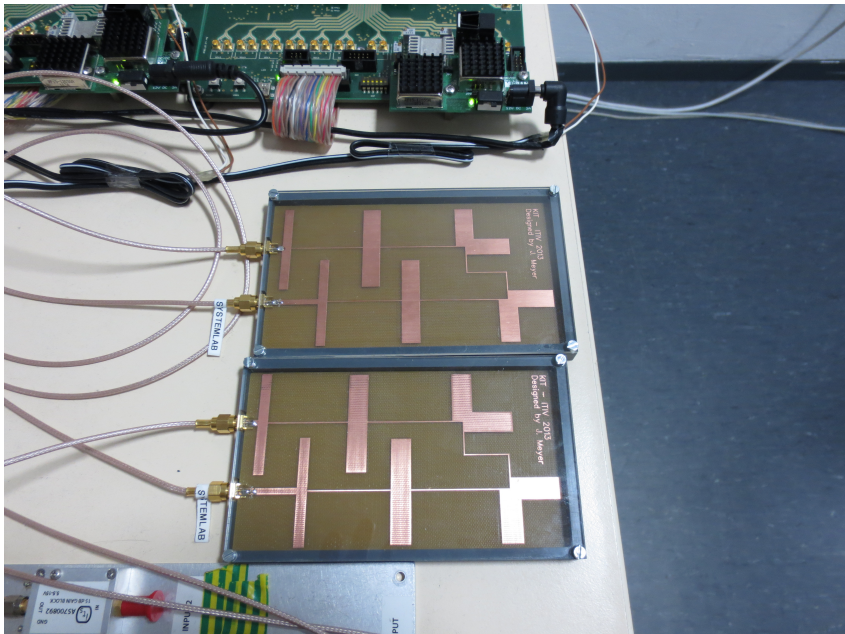


Abbildung 6.12.: Analoger Tiefpassfilter am ONU

## 7. Testergebnisse

In diesem Kapitel werden die in dieser Arbeit ermittelten Ergebnisse vorgestellt. Es sollte an dieser Stelle bereits erwähnt werden, dass das Gesamtsystem niemals komplett integriert aufgebaut und betrieben wurde. Der Grund dafür liegt nicht bei Konzeptions- oder Realisierungsproblemen einzelner FPGA- oder Elektronikkomponenten, sondern daran, dass leider bis zu Letzt lediglich ein FPGA Mainboard aus der Prototypenserie und damit nicht genügend Boards zum gleichzeitigen Betrieb eines OLT sowie einer oder gar mehrerer ONUs verfügbar waren. Aufgrund der Komplexität der Platine sowie den hohen Produktionskosten kam es immer wieder zu massiven Verzögerungen sowohl bei der Platinenproduktion als auch bei Bestückung und Test.

Unter Zuhilfenahme von AWGs und Oszilloskopen konnten jedoch trotzdem auch mit einem Board die wesentlichen Elemente des Signalverarbeitungssystems sequentiell in realer Hardware getestet werden. Auch das Zusammenspiel der einzelnen Komponenten der MAC Schicht wurde, ohne die OFDM Signalverarbeitung, verifiziert. Schließlich wurde parallel zur Erstellung der einzelnen Prozessierungsmodule auch eine Systemsimulation auf Basis von Modelsim aufgebaut, welche ein OLT mit zwei ONUs verbindet, indem das Verhalten mehrerer 10G Ethernet Signalgeneratoren, der verschiedenen AD bzw. DA Wandler sowie der Übertragungsstrecke möglichst realitätsnah simuliert wurde.

### 7.1. Komponententests

Bevor Systemtests, welche größere Teile des Signalverarbeitungssystems einschließen durchgeführt werden, muss zunächst die korrekte Funktion aller erforderlichen Komponenten einzeln verifiziert werden. Viele

Module realisieren für sich gesehen relativ einfache Funktionen, die sich gut mit einer Verhaltenssimulation in Modelsim verifizieren lassen. Da diese Module in der Regel ausschließlich in einer Taktdomäne arbeiten und ihre Umsetzung daher in realen FPGAs keine Probleme erwarten lässt, wurde bei den meisten einfachen Modulen auf einen dedizierten Test in realer FPGA Hardware verzichtet. Zu diesen Modulen gehören beispielsweise die NCOs, die Equalizer, Modulatoren und Demodulatoren oder auch das Phasentrackingsystem der Pilottöne. Der Test der 256 Punkte bzw. 32 Punkte FFT verlief etwas aufwändiger. Zwar arbeiten auch diese Module lediglich in einer Taktdomäne, ihre Funktion sowie auch ihr Ressourcenbedarf sind, verglichen mit vielen anderen Modulen, jedoch deutlich komplexer bzw. höher.

### 7.1.1. FFT Tests

Wie in Abschnitt 2.2.2 erläutert wurde, erfolgt bei der Berechnung einer FFT nach jeder Butterfly Stufe eine Multiplikation mit Drehfaktoren. Nach jeder Multiplikation muss das Zwischenergebnis gerundet werden, da ansonsten die Bitbreiten der Festkomma-Darstellung zu stark ansteigen würden. Auch die Wahl der Genauigkeit der Drehfaktoren (Anzahl der Nachkommastellen in dualer Repräsentation) geht selbstverständlich in die Qualität der FFT Berechnung ein.

Die Hardwareumsetzung der FFT wurde so realisiert, dass die Genauigkeit der Festkommadarstellung nach den Drehfaktormultiplikationen zum Synthesezeitpunkt mit Hilfe von Konfigurationsoptionen beliebig angepasst werden kann. Zur Verifikation der Berechnungsgenauigkeit der FFT Hardwarerealisierung wurde eine Referenzimplementierung in C, basierend auf Fließkommaberechnungen, verwendet. Die Ergebnisse mehrerer zufällig gewählter Eingangsvektoren, welche auch die Eckpunkte des verfügbaren Wertebereichs berücksichtigen, wurden in einer Modelsim Simulation der Hardwarerealisierung der FFT für verschiedene Konfigurationsoptionen verglichen. Nach mehreren Test wurde eine Konfiguration gewählt, welche gute Genauigkeit bei vertretbarem Ressourcenaufwand ermöglicht. Tabelle 7.1 listet die gewählte Konfiguration auf.

FFT Operation	Bitbreite
Eingangsvektor	6 Bit
Nach erster Radix-4 Berechnungsstufe	8 Bit
Nach erster Drehfaktor Multiplikationsstufe	11 Bit
Nach zweiter Radix-4 Berechnungsstufe	13 Bit
Nach zweiter Drehfaktor Multiplikationsstufe	13 Bit
Nach dritter Radix-4 Berechnungsstufe	15 Bit
Nach dritter Drehfaktor Multiplikationsstufe	15 Bit
Nach vierter Radix-4 Berechnungsstufe	17 Bit
Nach finaler Skalierung	14 Bit

Tabelle 7.1.: Gewählte Bitbreitenkonfiguration der 256 Punkte FFT

Abbildung 7.1 zeigt einen Vergleich zwischen einer mit Fließkommaarithmetik berechneten FFT und der Hardwarerealisierung mit der gewählten Konfiguration. Dabei ist zu beachten, dass der Eingangsvektor der Hardwarerealisierung, passend zu den Micram ADCs, lediglich sechs Bit breite Samples zulässt. Entsprechend kamen zur Ermittlung der Histogramme in Abbildung 7.1 trotz zufällig gewählter Eingangsvektoren lediglich Eingangswerte zum Einsatz, welche sich mit sechs Bit exakt, d.h. ohne Quantisierungsfehler, darstellen lassen, so dass ein direkter Vergleich der beiden FFT Implementierungen möglich ist. Erlaubt man stattdessen beliebige Eingangswerte, so dass der Quantisierungsfehler des sechs Bit AD Wandlers mit berücksichtigt wird, erhält man dagegen eine Fehlerverteilung wie in Abbildung 7.2 dargestellt. Man erkennt leicht, dass die geringe Bitbreite des AD Wandlers einen deutlich größeren Einfluss auf die Genauigkeit des Ergebnisses hat, als die FFT Berechnung selbst.

Da es sich bei der FFT um ein sehr ressourcenintensives Modul handelt, wurde es nach der Verifikation der grundsätzlichen Funktion und Berechnungsgenauigkeit schließlich noch als einzelnes Modul für den Ziel-FPGA synthetisiert, platziert und verdrahtet. So konnte überprüft werden, ob sich das Modul sinnvoll platzieren lässt und welche maximale Taktfrequenz erreicht wird. Die Tabellen 7.2 sowie 7.3 fassen die ermittelten Ergebnisse zusammen.

## 7. Testergebnisse

---

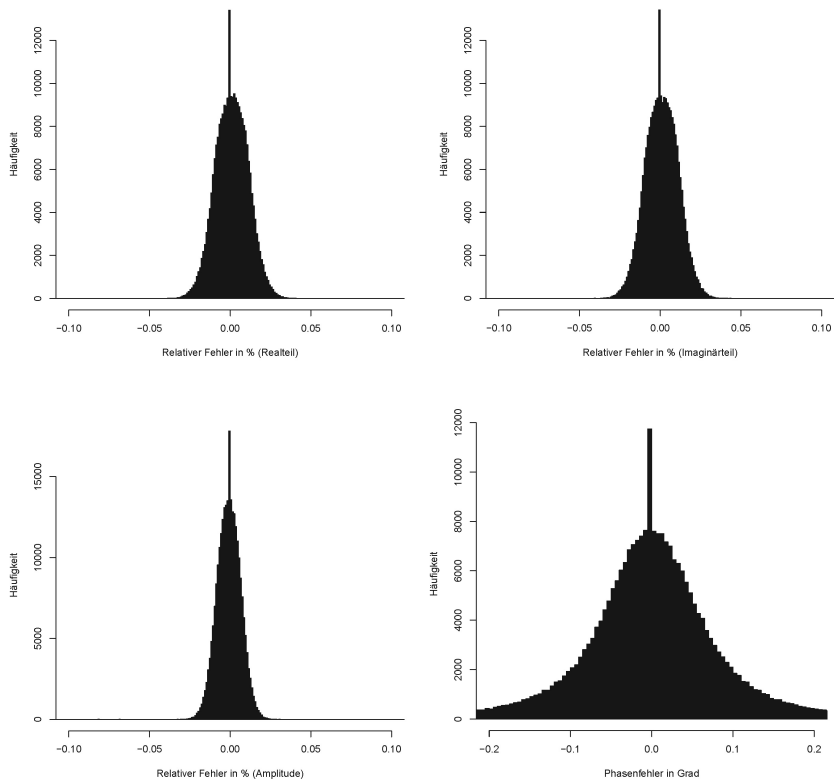


Abbildung 7.1.: Quantisierungsfehler der realisierten FFT für 1024 Transformationen bei zufälligen Eingangsvektoren ohne Eingangsquantisierungsfehler



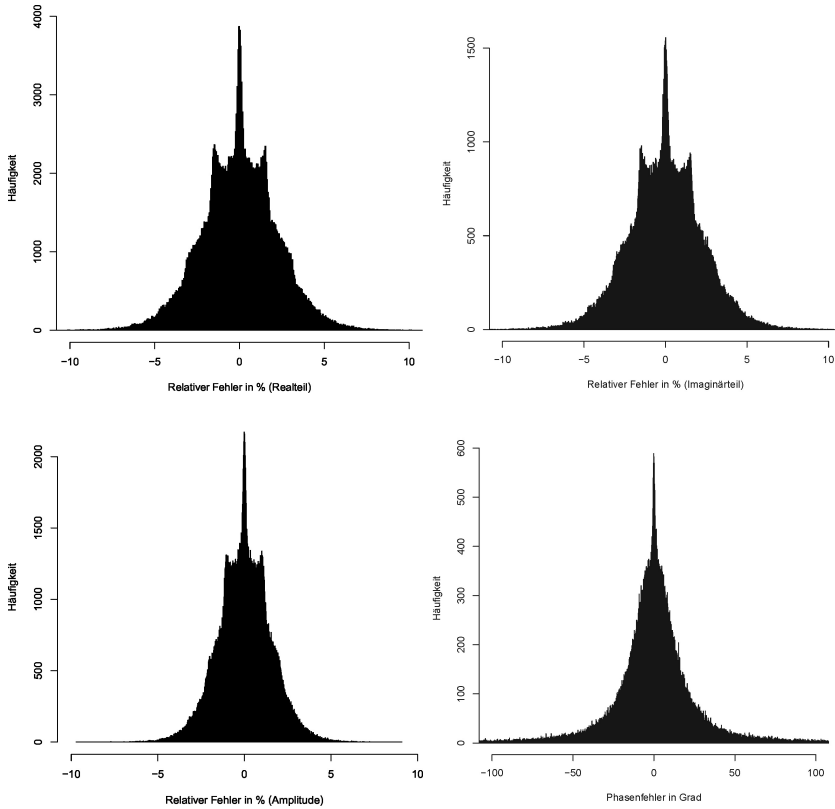


Abbildung 7.2.: Quantisierungsfehler der realisierten FFT für 1024 Transformationen bei zufälligen Eingangsvektoren mit Eingangsquantisierungsfehler

## 7. Testergebnisse

---

<b>Eigenschaft</b>	<b>Ergebnis</b>
Maximale Frequenz	561.7 MHz
Prozessierungsverzögerung	53 CLK
Anzahl genutzter Flip-Flops	191449
Anzahl genutzter LUTs	126599

Tabelle 7.2.: FFT 256 Synthese Ergebnisse

<b>Eigenschaft</b>	<b>Ergebnis</b>
Maximale Frequenz	248.3 MHz
Prozessierungsverzögerung	53 CLK
Anzahl genutzter Flip-Flops	191449
Anzahl genutzter LUTs	142481
Anzahl belegter Slices	48335
Anzahl ungenutzter Flip-Flop Paare	171863
Vollständig genutzte LUT-Flip-Flop Paare	130377

Tabelle 7.3.: FFT 256 Place & Route Ergebnisse

### 7.1.2. IO Interfaces

Bei dem in dieser Arbeit präsentierten Kommunikationssystem spielt die IO Leistung der eingesetzten FPGAs eine besonders große Rolle. Neben einem 10G Ethernet Interface wird auch Speicher mit einer Bandbreite von bis zu 25 GByte/s angebunden sowie verschiedene AD bzw. DA Wandler mit einem Durchsatz von bis zu 300 GBit/s. Damit all diese Schnittstellen zuverlässig arbeiten, mussten eine Menge Hardwaretests und Optimierungen durchgeführt werden. Der Einsatz von Simulationssoftware wie Modelsim ist an dieser Stelle nur in sehr begrenztem Umfang möglich, da viele Interfaceschaltungen spezielle FPGA Komponenten beinhalten, die sich gar nicht, oder nur sehr ungenau simulieren lassen. Weiterhin kommen bei komplexen, insbesondere bei MGTs basierten Schnittstellen oft viele verschiedene Taktdomänen zum Einsatz, welche häufig auch von unterschiedlichen Oszillatoren gespeist werden. Diese lassen sich Prinzip bedingt gar nicht oder nur sehr ungenau simulieren. Die Schnitt-

stellentests fanden daher primär direkt auf der Ziel-FPGA Hardware und zusammen mit den anzubindenden Hardwarekomponenten, wie z.B. den Micram Wandlern, statt.

Die Realisierung der DDR3 Speicherinterfaces ging relativ einfach und schnell vonstatten. Das lag vor allem daran, dass während des Designs des FPGA Mainboards sehr genau auf eine korrekte und von Xilinx freigegebene Pinbelegung des Speicherinterfaces geachtet wurde. Somit konnten die von Xilinx bereitgestellten Speicherinterface-Cores direkt und ohne große Veränderungen eingesetzt werden.

Die Anbindung des 10G Ethernet Interfaces gestaltete sich ebenfalls verhältnismäßig einfach. Auch hier konnte aufgrund der Standardisierung der Schnittstelle auf vorhandene Interface-Cores von Xilinx bzw. Opencores zugegriffen werden. Die Hauptschwierigkeit bestand in der physikalischen Anbindung eines SFP+ Transceiver Moduls an einen Test FPGA, während das eigentliche Mainboard mit integrierten SFP+ Sockeln noch nicht verfügbar war. Zur Einsparung von Kosten wurde lediglich einfaches Platinenmaterial verwendet, welches jedoch den hohen auftretenden Frequenzen von über 5 GHz nicht ganz gewachsen war und entsprechend hohe Dämpfungswerte sowie Impedanzprobleme verursachte. Die Übertragung von Ethernetpaketen war somit recht unzuverlässig, sie reichte jedoch für die Tests bis zum Eintreffen des eigentlichen Mainboards aus. Ein Vorteil dieser unzuverlässigen Übertragung war jedoch, dass das *Error Filter* Modul der MAC Layer (vgl. Abbildung 4.18) ausgiebig getestet werden konnte. Abbildung 7.3 zeigt ein Bild des eingesetzten SFP+ Adapterboards.

Der mit Abstand größte Aufwand bestand in der Anbindung der 25 GSa/s sowie der 3.125 GSa/s AD und DA Wandler. Hier existierten weder vorgefertigte Interface-Cores noch ein klares Konzept zur Anbindung der Bausteine. Micram sieht für die Kommunikation mit ihren 25 GSa/s Wandlern keinerlei Linecode vor, welcher die Synchronisation der einzelnen MGLs unterstützen könnte. Stattdessen werden die Datenbits völlig uncodiert bzw. mit einer PRBS XOR-verknüpft übertragen. Die Herausforderung bzgl. Test und Inbetriebnahme des Interfaces bestand im Wesentlichen darin, die 48 parallel arbeitenden MGLs, welche für zwei Wandler benötigt werden, zueinander zu synchronisieren. Dabei konnte jedoch keinerlei Simulation eingesetzt werden, da das Verhalten

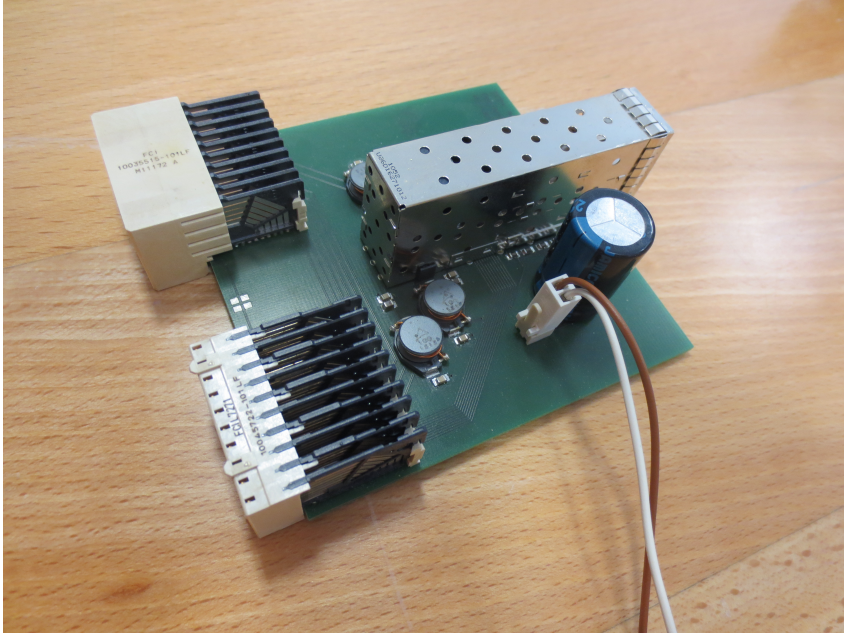


Abbildung 7.3.: SFP+ Interface Adapter

der Micram Wandler nicht genau bekannt und auch die Dokumentation der Bausteine zum Teil fehlerhaft war. Stattdessen kamen direkt an der FPGA Schnittstelle der GTX Transceiver mehrere Instanzen von ChipScope zum Einsatz. So konnte die Software des den Synchronisationsvorgang steuernden Prozessorsystems direkt, basierend auf dem Verhalten der echten Hardwarerealisierung, optimiert werden.

Auch die Anbindung der 3.125 GSa/s Wandler bedeutete größeren Aufwand. Zwar konnte hier aufgrund der zusätzlich eingesetzten FPGAs auf den Wandlerplatinen der Linecode der MGLs weitgehend frei bestimmt und mit 8B10B ein selbst synchronisierender Code eingesetzt werden, jedoch mussten auch hier die einzelnen Links untereinander synchronisiert werden, damit sich am Ende ein Datenkanal mit der erforderlichen Bandbreite ergibt. Auch bei diesem Interface wurden die meisten Tests

und Optimierungen mit Hilfe von ChipScope direkt an realer Hardware durchgeführt.

Dieses Vorgehen ist recht mühsam und langwierig, denn eine Änderung der Konfiguration von ChipScope zieht oft lange Toollaufzeiten im Bereich von vielen Stunden nach sich. Auch die Triggerung auf das gewünschte Ereignis kann recht kompliziert sein und gerade bei sehr selten auftretenden Ereignissen gehört häufig auch etwas Glück dazu. Ist ein Problem jedoch einmal identifiziert und behoben, ist gleichzeitig sichergestellt, dass die betroffene Schnittstelle auch wirklich korrekt arbeitet.

Auch während der Realisierung und den Tests der Wandlerinterfaces war das spätere FPGA Mainboard noch nicht verfügbar, so dass auch hier auf provisorische Testaufbauten zurückgegriffen werden musste wie Abbildung 7.4 zeigt.

## 7.2. Systemsimulation

Die Hauptaufgabe der Systemsimulation ist die simulative Verbindung von Vermittlungsstelle und Teilnehmeranschlüssen. Dazu wird sowohl das Verhalten der AD bzw. DA Wandler simuliert, als auch die eigentliche analoge elektro-optische Übertragungsstrecke. Ohne diese Streckensimulation wäre eine direkte Verbindung von OLT und ONU gar nicht möglich, da sich die eingesetzten Sampleraten unterscheiden und außerdem Frequenzverschiebung und -selektion für die korrekte Entnahme einer spektralen Gruppe aus dem OFDMA Signal des OLT notwendig sind. Da die Simulation mit den selben Very High Speed Description Language (VHDL) Modulen arbeitet, welche auch bei der Synthese für die FPGAs zum Einsatz kommen, ergibt sich ein sehr akkurates Simulationsergebnis, welches tiefe Einblicke in das Verhalten der eingebundenen Funktionsmodule erlaubt während diese in einer realistischen Umgebung betrieben werden.

Eine weitere sehr wichtige Anwendung der Systemsimulation ist die Generierung und Analyse von Signaldatenströmen für den Test einzelnen Netzwerkkomponenten in Hardware. Für die Signalgenerierung wird ein von OLT oder ONU simulativ erzeugtes OFDMA Signal vor oder

## 7. Testergebnisse

---

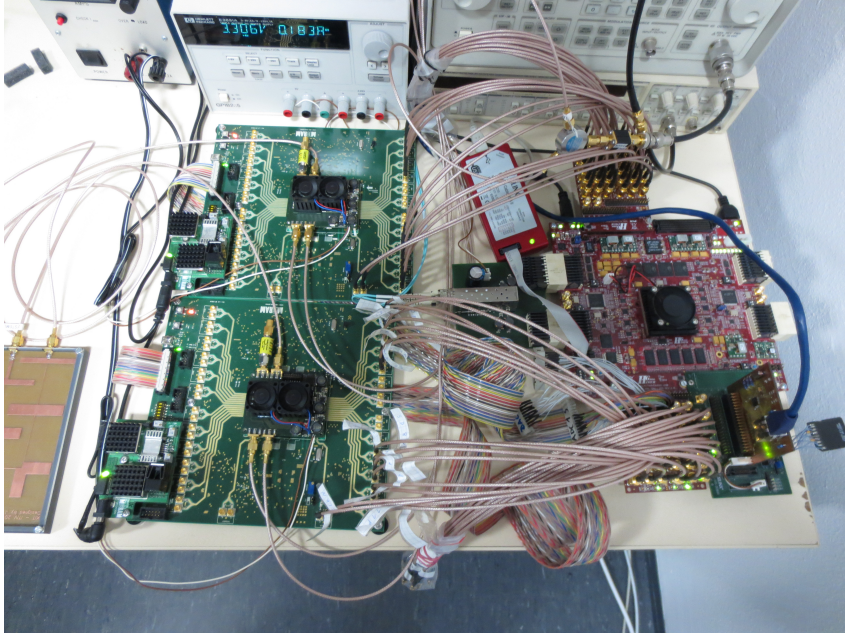


Abbildung 7.4.: Testaufbau zur Entwicklung der Wandlerinterfaces

nach der Streckensimulation ausgekoppelt und mit Hilfe eines AWGs einer realen Umsetzung eines Teil des Netzwerks zugeführt um dessen Funktion zu testen. Alternativ wird bei der Signalanalyse ein von realen Netzwerkkomponenten erzeugtes Signal mit Hilfe eines Oszilloskops aufgenommen und anschließend mit Hilfe der Systemsimulation verarbeitet. Auf diese Weise ist es beispielsweise einfach möglich, die Auswirkungen von Änderungen an Empfängerparametern auf reale Signale zu untersuchen. Abbildung 7.5 zeigt ein Blockschaltbild der aufgebauten Simulationsstrecke für den Downlink.

Zu Beginn wird der komplexe, parallele Samplevektor der IQ Signalisierung vom Transmitter des OLT entgegengenommen. Die Simulation der DA Wandler generiert daraus ein serielles Signal im Fließkommaformat. An dieser Stelle besitzt es, wie die realen Micram DACs eine Samplerate von 25 GSa/s bei einer sechs Bit Quantisierung. Bevor nachfolgende

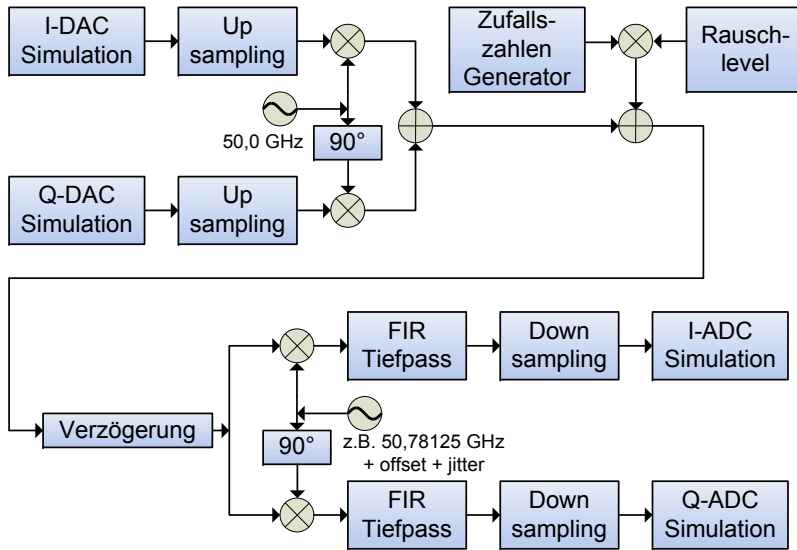


Abbildung 7.5.: Simulationsstrecke des Downlinks

Operationen wie ein IQ Mischvorgang oder Frequenzverschiebungen erfolgen können, muss zunächst die Samplerate erhöht werden. Da die Bandbreite des komplexen Basisbandsignals des OLT ebenfalls 25 GHz beträgt, wäre aufgrund des Abtasttheorems eine Samplerate von 50 GSa/s ausreichend für die Darstellung des reellen Signal nach einem IQ Mischer. Um eine für den Benutzer bessere optische Anschauungsqualität der Signale zu erzielen, wurde in der Simulation jedoch eine Abtastfrequenz von 500 GSa/s gewählt.

Die Abtastfrequenz des Ausgangssignals der beiden DA Wandler wird nun zunächst mit Hilfe einer Sinc-Interpolation von 25 GSa/s auf 500 GSa/s erhöht. Anschließend erfolgt der IQ Mischvorgang mit einer Trägerfrequenz von 50 GHz. Das Mischen der Signale mit dem Träger kann aufgrund der hohen Überabtastung einfach mit Hilfe einer Multiplikation durchgeführt werden. Das Trägersignal selbst wird mit Hilfe eines NCO erzeugt, welcher direkt bei 500 GSa/s taktet, so dass keine Konvertierung der Samplerate benötigt wird. Das Ergebnis des Mischvorgangs stellt das

OLT Sendesignal dar, wie es in der Realität auch genutzt wird, um die Amplitude des Lasersignals im Downlink zu modulieren.

Während der Übermittlung des Downlinksignals im Feld wirken verschiedene Störeinflüsse darauf ein. Hier spielt vor allem die Signaldämpfung eine wichtige Rolle, da sie durch nachfolgende Verstärkung zu Rauschen und damit zu einem schlechteren SNR führt. Ein weiterer, nicht zu unterschätzender Einfluss des Downlinks ist eine Signalverzögerung, die bei der anvisierten Gesamtnetzlänge von 100 km durchaus einige 10  $\mu$ s erreichen kann und bei der Synchronisation des Netzwerks entsprechend berücksichtigt werden muss. Aus diesem Grund können für die Streckensimulation der Rauschanteil sowie die Verzögerungszeit individuell eingestellt werden, um den Empfang des Downlinksignals an einer ONU in unterschiedlichen Empfangssituationen testen zu können.

Nachdem das Signal wie gewünscht *degradiert* wurde, erfolgt der eigentliche Empfang durch eine ONU. Dazu erfolgt das Heruntermischen des Signals ins komplexe Basisband mittels eines IQ Demodulators. Die spektrale Gruppe wird dabei durch die Wahl einer geeigneten Trägerfrequenz selektiert. Für den Zugriff auf spektrale Gruppe 7 wird beispielsweise eine Trägerfrequenz von 50.78125 GHz verwendet, da diese um +0.78125 GHz von der gewählten Trägerfrequenz am OLT entfernt liegt, wie aus Abbildung 4.5 leicht ersichtlich ist. Wie auch beim Hochmischen des Basisbandsignals des OLT wird der Träger von einem NCO generiert. Hier besteht jedoch in der Simulation zusätzlich die Möglichkeit, einen Frequenzoffset sowie Frequenzjitter vorzugeben, um ein realistisches Verhalten des Frequenzoffsets zwischen den Lokaloszillatoren von OLT und ONU zu testen. Im nächsten Schritt muss das Basisbandsignal Tiefpassgefiltert werden, bevor seine Samplerate herabgesetzt wird, da es noch sehr viele Mischprodukte der restlichen spektralen Gruppen in oberen Frequenzanteilen enthält. Die beiden Tiefpassfilter für den I und den Q Anteil des Basisbandsignals sind in der Simulation in Form eines Finite Impulse Response (FIR) Filters realisiert. Das Abfallen der Filterflanke beginnt bei 781.25 MHz und ab der doppelten Frequenz von 1.5625 GHz werden Signalanteile schließlich vollständig unterdrückt. Da die Wandler der ONU mit 3.125 GSa/s arbeiten, kann das gefilterte Signal problemlos digitalisiert werden. Die Signalanteile der benachbarten spektralen Gruppen im Bereich zwischen 781.25 MHz und 1.5625 GHz werden nachträglich mit Hilfe der im ONU Empfänger arbeitenden FFT entfernt



(siehe Abschnitt 4.2). Nach der Tiefpassfilterung kann das Signal schließlich einfach mit 3.125 GSa/s unter-abgetastet werden und mit Hilfe der ADC Simulation in einen parallelen Samplevektor zur Weiterverarbeitung in einer ONU umgesetzt werden. Die entgegengesetzte Strecke des Uplinks arbeitet prinzipiell identisch mit dem Unterschied, dass hier das Sendesignal der ONU interpoliert wird und die Tiefpassfilterung und Unterabtastung des Signals am OLT Empfänger erfolgt.

Prinzipiell kann an jeder Stelle der Simulation ein Signal aus- oder eingekoppelt werden, um reale Hardware in den Simulationsvorgang einzubinden. Der Austausch der Simulationsdaten zwischen Modelsim und AWG bzw. Oszilloskop erfolgt dabei mit Hilfe von einfachen Textdateien, in denen die Werte der Samples zeilenweise abgelegt sind. Mit Unterstützung der Simulation ließen sich die meisten Signalverarbeitungsfunktionen in realer Hardware testen. Einschränkungen existierten lediglich beim Test der vollständigen Synchronisation des Uplinks, da hierzu in Echtzeit eine bidirektionale Kommunikation zwischen OLT und ONU mit Hilfe von Kontrollnachrichten erfolgen muss, wozu vorberechnete Signalverläufe aus der Simulation nicht geeignet sind. Abbildung 7.6 zeigt beispielsweise einen AWG, welcher das von einer OLT Simulation generierte IQ Basisbandsignal des Downlinks für einen Test mit einem ONU Empfänger in realer Hardware bereitstellt. An der dargestellten Wellenform können die Rahmengrenzen des OFDM Signals deutlich abgelesen werden. Da nur die Subträger einer ONU moduliert wurden, ist die Amplitude im Datenbereich deutlich geringer als in den Synchronisations- bzw. Kontrolldatenbereichen, in denen alle Subträger aktiv sind.

Die Ergebnisse der Systemsimulation sind sehr vielversprechend. Es konnten trotz verrauschtem Übertragungskanal und aktiviertem Frequenzoffset und -jitter der analogen ONU Simulation eine vollständige Synchronisation des Downlinks und Uplinks erzielt und Ethernetpakete erfolgreich transferiert werden.

## 7.3. Systemtests

Für die Durchführung der Systemtests wurden größere Teile des Kommunikationssystems in FPGA Hardware real aufgebaut und getestet.

## 7. Testergebnisse

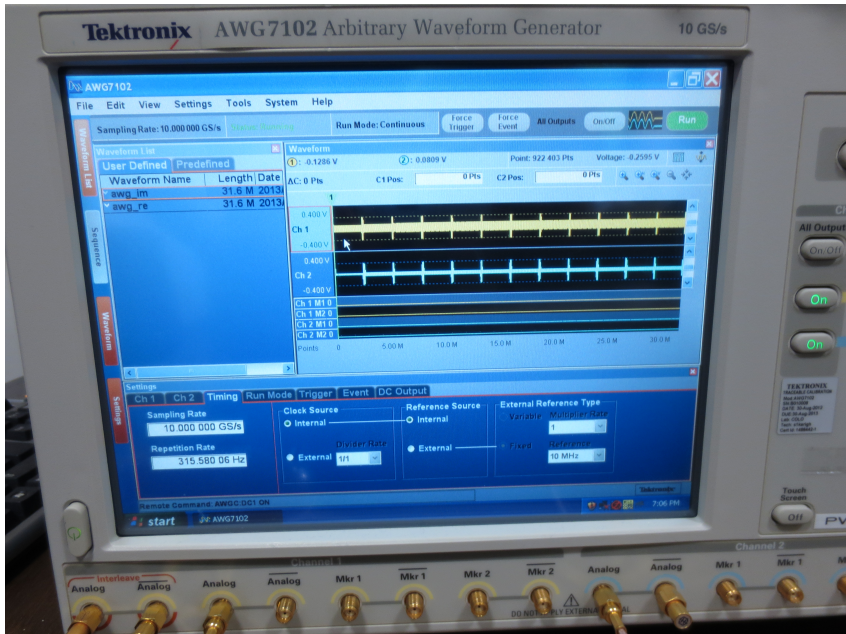


Abbildung 7.6.: Ausgabe eines simulierten Downlinksignals durch einen AWG

### 7.3.1. Packetdatenverarbeitung

Zunächst erfolgte der Test der 10G Ethernet Anbindung und der MAC Layer, welche für die Priorisierung, Absicherung und Aufteilung des Paketdatenstroms auf die einzelnen Subträger des OFDMA Systems verantwortlich ist (siehe Abschnitt 4.3.3). Abbildung 7.7 zeigt die dazu gewählte Toplevel-Konfiguration des FPGAs. Dabei handelt es sich im Prinzip um das in Abbildung 4.19 dargestellte FPGA-Gesamtsystem, wobei die OFDM Signalverarbeitung und die analogen Interfaces entfernt wurden. Stattdessen wird der Ausgang der MAC Layer zur Modulation der Subträger direkt auf den entsprechenden Eingang zurück geführt. Ein Ethernetpaket, welches vom 10G Interface empfangen wird und die MAC Layer passiert hat, wird somit über das gleiche 10G Interface wieder

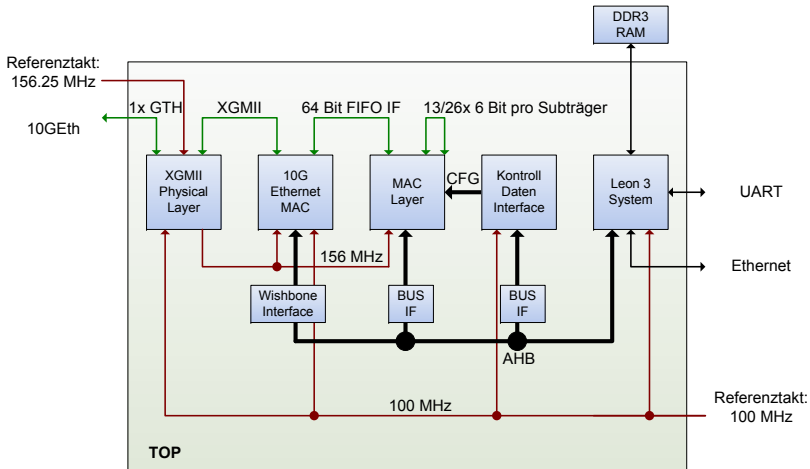


Abbildung 7.7.: Test der Ethernetanbindung

ausgesendet. Auf diese Weise konnte die komplette Paketverarbeitung im realen Betrieb detailliert getestet werden.

### 7.3.2. OFDM Signalverarbeitung

Die mit Abstand kritischste Komponente des Systems ist der Downlinkempfänger einer ONU, denn hier finden sämtliche komplexen Synchronisationsaufgaben statt. Zum Test des Empfängers wurde eine vollständige ONU, bestehend aus einem analogen Frontend Board (vgl. Abschnitt 6.5) sowie einem FPGA mit einer vollständigen ONU Implementierung eingesetzt. Aufgrund fehlender Hardware musste das OLT mit Hilfe eines AWG emuliert werden, so dass es leider nicht möglich war, Ethernetpakete in Echtzeit am OLT in den Downlink einzuspeisen. Der für den AWG vorbereitete Datensatz wurde mit Hilfe einer Simulation des OLT generiert und enthielt zwölf Frames in deren Datenbereichen unterschiedliche Ethernetpakete eingebettet waren. Zunächst wurde die korrekte Funktion des Synchronisationsmoduls überprüft, anschließend konnte auch der korrekte Empfang der Ethernetpakete durch die MAC Layer der ONU

## 7. Testergebnisse

```
General receiver process values
-----
Sync sequence based correction value for LO: -0001528603
Sync sequence based correction value for SO: +0000000000
CPE based correction value for LO: -0000042165
DPE based correction value for SO: +0000000000

Local oscillator frequency error: -488.3 KHz
Sampling oscillator frequency error: +3125000.0 KHz

Number of detected synchronization pulses: 0000006648
Receiver locked state: 1

Common mode phase error: -0000033756
Differential mode phase error: +0000000521

Channel 00 equalizer: Amplitude: +0.00000 Phase: +nan deg
Channel 01 equalizer: Amplitude: +0.77973 Phase: -48.12213 deg
Channel 02 equalizer: Amplitude: +1.34334 Phase: -132.84947 deg
Channel 03 equalizer: Amplitude: +0.91379 Phase: +168.06562 deg
Channel 04 equalizer: Amplitude: +1.12547 Phase: +85.54601 deg
Channel 05 equalizer: Amplitude: +1.17757 Phase: +54.97878 deg
Channel 06 equalizer: Amplitude: +1.07714 Phase: -6.45492 deg
Channel 07 equalizer: Amplitude: +1.00191 Phase: -54.14097 deg
Channel 08 equalizer: Amplitude: +1.07442 Phase: -82.19027 deg
Channel 09 equalizer: Amplitude: +0.88370 Phase: -32.52218 deg
Channel 10 equalizer: Amplitude: +0.75151 Phase: -76.89377 deg
Channel 11 equalizer: Amplitude: +0.87659 Phase: -120.86524 deg
Channel 12 equalizer: Amplitude: +0.83219 Phase: +177.00665 deg
Channel 13 equalizer: Amplitude: +0.65965 Phase: +108.91777 deg
Channel 14 equalizer: Amplitude: +0.52801 Phase: +65.75776 deg
Channel 15 equalizer: Amplitude: +0.70317 Phase: +15.75412 deg

Control message reception
-----
Ctrl Message received: 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
█
```

Abbildung 7.8.: Kontrollfenster einer ONU

und deren Ausgabe über die 10G Schnittstelle verifiziert werden. Abbildung 7.8 zeigt die dabei zum Einsatz gekommene Kontrollanwendung in einer Linux-Konsole.

Parallel dazu wurde das FPGA System der getesteten ONU simuliert, um tiefere Einblicke in das System während seines Betriebs zu erhalten. Dazu wurde der simulierte Datenstrom des OLT einmal durch die Systemsimulation und einmal durch das reale optische Netzwerk geleitet und das an der ONU ankommende Signal aufgezeichnet und in die Simulation geleitet. Einige Simulationsergebnisse sind im Folgenden dargestellt.

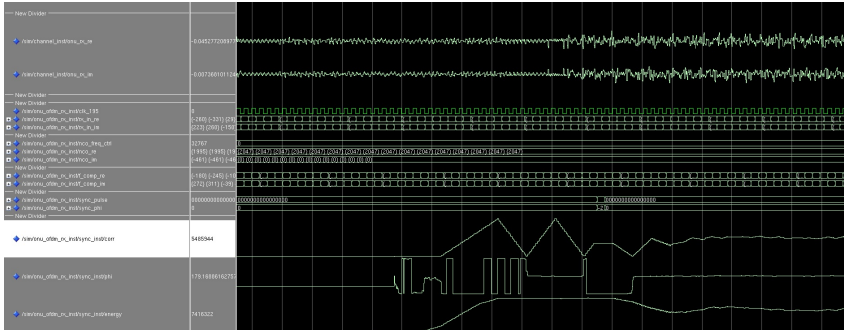


Abbildung 7.9.: Korrelations- und Energiesignal während dem Empfang einer ungestörten Synchronisationssequenz

Abbildung 7.9 zeigt die Ausgabe der Synchronisationseinheit des ONU beim Empfang eines ungestörten Signals. Es sind zwei klare Korrelationspitzen zu erkennen (Signal *corr*), welche von den zwei identischen Hälften des Synchronisationssignals herrühren. Das Energiesignal (Signal *energy*) bleibt während der zweiten Spitze, welche zur Auswertung des Synchronisationszeitpunktes herangezogen wird, weitgehend konstant. Auch der ermittelte Phasenfehler (Signal *phi*), welcher zur Korrektur des Lokaloszillator-Frequenzoffsets verwendet wird, bleibt hier konstant.

In Abbildung 7.10 ist das Auftreten eines konstanten, differentiellen Phasenfehlers zu erkennen (Signal *pet\_dpe*). Die Phase driftet allmählich nach unten weg, da sie in diesem Fall nicht korrigiert wird. Im Gegensatz dazu ist in Abbildung 7.11 die aktive Korrektur eines gemeinsamen Phasenfehlers (Signal *pet\_cpe*) mit Hilfe eines PID Controllers dargestellt. Zu Beginn der Übertragung schwingt der Regler noch etwas, was an dem leicht sinusförmigen Verlauf des Phasenfehlers erkennbar ist. Zum Ende hin stabilisiert sich der Regelkreis jedoch und der gemeinsame Phasenfehler wird zuverlässig zu Null geregelt.

## 7. Testergebnisse

---

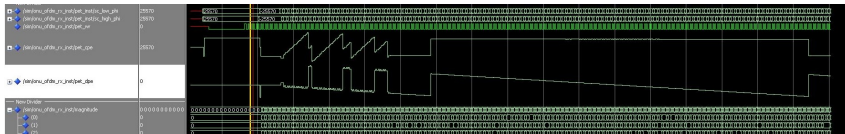


Abbildung 7.10.: Differenzieller Phasenfehler bei einem Frequenzoffset des Sampleoszillators

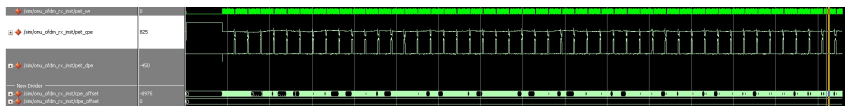


Abbildung 7.11.: Ausregelung eines gemeinsamen Phasenfehlers

# 8. Schlussfolgerung und Ausblick

## 8.1. Zusammenfassung und Bewertung der Ergebnisse

In den vorherigen Kapiteln wurde ein hochperformantes Kommunikationssystem auf OFDMA Basis für den Einsatz in optischen Zugangsnetzwerken vorgestellt. Dabei lag der Fokus der Arbeit auf der Entwicklung und Umsetzung eines Gesamtkonzepts, welches neben der Umsetzung der reinen OFDMA Signalisierung auch die Einbettung und Übermittlung von realen Verkehrsdatenströmen in Form von 10G Ethernet Paketen betrachtet. Damit unterscheidet es sich stark von anderen Arbeiten im Bereich der Hochgeschwindigkeitskommunikation, welche ebenfalls auf komplexen, modernen Modulationsverfahren beruhen. Es stellt daher eins der ersten Systeme dar, wenn nicht das erste System überhaupt, welches beim Einsatz eines modernen und komplexen Modulationsverfahrens eine vollumfängliche Kommunikation in der Geschwindigkeitsklasse von 50 GBit/s abdeckt.

Alle wesentlichen Teile des Systems wurden von Grund auf konzipiert, realisiert und getestet. Auch wenn ein kompletter Systemtest aufgrund fehlender Hardwarekomponenten leider ausbleiben musste, konnte doch die korrekte Funktion sämtlicher entscheidender Systemteile durch eine Kombination von Simulation, Emulation und realen Tests verifiziert werden.

Eine wesentliche Erkenntnis aus dem Aufbau eines solchen Gesamtsystems ist, dass neben der OFDMA Signalverarbeitung auch noch weitere kritische Komponenten bzgl. Performanz und Ressourcenverbrauch existieren, wie z.B. ein immer benötigter FEC Decoder. Auch die Limitierung

der IO Bandbreite und die verfügbaren Schnittstellen moderner Hardware stellen immer noch eine Herausforderungen dar.

### 8.2. Übertragbarkeit auf andere Kommunikationssysteme

Als Basis für die Demonstration eines modernen Hochgeschwindigkeitskommunikationssystems diente in dieser Arbeit ein orthogonales Mehrträgerverfahren. Praktisch alle modernen Kommunikationsstandards wie beispielsweise DVB-T, LTE, WLAN oder auch DSL basieren auf einer Form von OFDM. Trotzdem stellt sich die Frage, in wie weit sich die ermittelten Ergebnisse auf andere Kommunikationssysteme und Modulationsformate übertragen lassen.

Wie in Kapitel 2 erläutert wurde, können digitale Übertragungsverfahren grundsätzlich in Ein- und Mehrträgerverfahren unterteilt werden. Wobei für die Mehrträgerverfahren zusätzlich seit neuestem MIMO Verfahren beliebt werden. Die grundlegenden Signalverarbeitungsblöcke auf denen all diese Verfahren beruhen sind jedoch im Wesentlichen identisch. Auch wenn beispielsweise in einem Empfänger für ein Einträgerverfahren nicht unbedingt eine Transformation vom Zeit- in den Frequenzbereich benötigt wird, kommt eine FFT hier trotzdem häufig als Subkomponente anderer Signalverarbeitungsblöcke, wie beispielsweise einer Filterbank, zum Einsatz. Auch andere in dieser Arbeit benötigte Blöcke wie z.B. eine Autokorrelation, NCOs oder Modulatoren und Demulatoren werden in den allermeisten Systemen eingesetzt.

OFDM-Erweiterungen wie beispielsweise das Single-Carrier FDMA (SC-FDMA) Verfahren erlauben die Reduktion des PAPR des Systems und beheben damit einen wesentlichen Nachteil von OFDM. Dazu werden die komplexen Symbole der Modulatoren vor ihrer Verteilung auf die Subträger durch eine zusätzliche FFT geleitet. D.h. auch für diese Erweiterung werden keine neuen Signalverarbeitungsbestandteile benötigt. Eine Ausnahme bildet lediglich die MIMO Technologie. Hier werden zusätzlich zu den typischen OFDM Komponenten rechenintensive Ma-



trixoperationen zur Auflösung der einzelnen räumlichen Signalströme benötigt.

Ein weiterer wichtiger Teil dieser Arbeit war die praktische Realisierung einer Hardwarebasis, die es zunächst einmal überhaupt erlaubt Samples mit der benötigten Abtastrate zu erfassen bzw. zu generieren. Dazu musste, speziell was die IO Schnittstellen der verwendeten Hardware angeht, einige Vorarbeit geleistet werden, welche selbstverständlich ebenfalls von jedem anderen digitalen Kommunikationssystem identisch wiederverwendet werden kann.

Angesichts der heute existierenden vielfältigen Verfahren zur bandbreiteneffizienten Informationsübermittlung deckt die in dieser Arbeit gewählte Realisierung daher einen sehr großen Bereich ab, so dass auch viele Systeme, die auf deutlich anderen Modulationsverfahren basieren von den gemachten Erfahrungen profitieren können.

Schließlich soll an dieser Stelle noch angemerkt werden, dass auch die hier entwickelten Komponenten in den allermeisten Parametern zur Synthesezeit sehr frei konfigurierbar sind. Somit lassen sich verhältnismäßig einfach auch Systeme mit völlig anderem Kommunikationsverhalten aufbauen, in dem die vorhandenen Komponenten neu parametrisiert und verschaltet werden. Eine Basis aus FPGAs wie sie für diese Realisierung verwendet wurde, ist dabei sehr vorteilhaft. Da einfach ein neuer Bitstrom geladen werden kann, lässt sich das Kommunikationssystem ähnlich einem Software Defined Radio (SDR) sehr einfach und schnell umkonfigurieren.

## 8.3. Ausblick

Zwei Themen, welche während dieser Arbeit ausgespart wurden, sind der Energieverbrauch des Systems und Mechanismen zur Verschlüsselung der übermittelten Daten.

Die Leistungsaufnahme der aktuellen Realisierung liegt mit praktisch gemessenen 100W relativ hoch. Interessanterweise wird dabei die meiste Leistung nicht von der eigentlichen FPGA Logik benötigt, sondern durch die eingesetzten DA und AD Wandler sowie deren Anbindung über

MGLs an die FPGAs wie in Abschnitt 6.2 erläutert wurde. Während dieser Wert für das OLT der Vermittlungsstelle vertretbar scheint, muss der Leistungsbedarf der ONUs sicherlich für einen praktischen Einsatz gesenkt werden. Bedenkt man, dass es sich bei den am ONU eingesetzten Wandlern um Prototypen mit eigenen FPGAs handelt, existiert hier noch starkes Einsparpotential. Beispielsweise könnten die Wandler bei einer hochintegrierten Version einer ONU mit auf den eigentlichen Chip zur Signalverarbeitung integriert werden. Diese Lösung wäre sicherlich die effektivste, da so sämtliche Kosten und die Leistungsaufnahme für die IO-Schnittstellen der Wandler komplett entfallen. Aber auch für eine rein digitale Umsetzung der Signalverarbeitung, beispielsweise in Form eines Standardzellendesigns, könnte die Bereitstellung passender DDR Schnittstellen für die Wandler zusätzliche FPGAs und MGLs und damit einen Großteil der Leistungsaufnahme einsparen.

In dem in dieser Arbeit vorgestellten System ist durch die P2MP Architektur des Netzwerks, zumindest theoretisch, im Downlink sämtlicher Datenverkehr aller Teilnehmer an jedem Anschluss abgreifbar. In der Praxis mögen weit vom OLT entfernte Anschlüsse aufgrund eines schlechteren SNR eventuell nicht in der Lage sein, eine höherwertige Modulation anderer Teilnehmer korrekt zu demodulieren. Auch ist der Aufbau eines Abhörgeräts bei den genutzten hohen Datenraten und Frequenzen sicherlich nicht trivial. Trotzdem erfordert ein solches System im praktischen Einsatz mit geteiltem Medium eine Verschlüsselung der Daten. Dabei besteht die Herausforderung nicht nur in der Konzeption geeigneter Ver- und Entschlüsselungsalgorithmen und deren hochperformanten Realisierung zur Echtzeitverarbeitung der anfallenden Datenströme. Zusätzlich muss auch ein Sicherheitskonzept mit Schlüsselmanagement und Nutzerauthentifizierung entwickelt werden. Wie schwierig und fehleranfällig diese Aufgabe sein kann, zeigen immer wieder spektakuläre Einbrüche in Kommunikationsinfrastrukturen.

Bezüglich des vorgestellten, bestehenden Systems existieren bereits zum jetzigen Zeitpunkt potentielle Nachfolgetechnologien. Beispielsweise werden auch die großen Virtex-7 FPGAs langsam lieferbar und könnten die Virtex-6 Technologie ersetzen und so noch mehr Ressourcen und Rechenleistung bereitstellen. Das wäre vor allem am OLT interessant, da so möglicherweise weniger FPGAs an dieser Stelle benötigt würden. Jedoch ändert sich dadurch nichts am prinzipiellen Vorgehen bei der Entwick-

lung und Umsetzung des Kommunikationssystems. Insbesondere der massive Einsatz von MGLs lässt sich in experimentellen Systemen, ohne hochintegrierte ASICs wie sie erst kurz vor einer Produktreife zum Einsatz kommen, in näherer Zukunft wohl nicht vermeiden.



# A. Anhang



# Abbildungsverzeichnis

2.1. QAM-16 Konstellationsdiagramm . . . . .	14
2.2. Momentanwert einer Sinusschwingung in komplexer Darstellung . . . . .	15
2.3. IQ Modulator . . . . .	17
2.4. OFDM Träger im Frequenzbereich . . . . .	21
2.5. Blockdiagramm eines stark vereinfachten OFDM Senders .	22
2.6. Blockdiagramm eines stark vereinfachten OFDM Empfängers . . . . .	22
2.7. FFT Radix-2 Butterfly . . . . .	24
2.8. FFT Radix-4 Butterfly . . . . .	24
2.9. 16 Punkt FFT in Radix 4 Struktur . . . . .	25
2.10. Vergleich zwischen DIT und DIF FFT . . . . .	26
2.11. Blockschaltbild eines NCO . . . . .	30
2.12. Um DDS Komponenten erweiterter NCO . . . . .	32
2.13. Blockschaltbild einer PLL . . . . .	33
2.14. Prinzipieller Aufbau eines PID-Reglers . . . . .	36
2.15. Vereinfachtes Blockdiagramm eines MGT . . . . .	40
4.1. Topologische Struktur eines optischen Zugangsnetzwerks	57
4.2. Optisches Testübertragungsnetz . . . . .	58
4.3. Analoges Tiefpassfilter am ONU . . . . .	69
4.4. Spektraler Aufbau des OFDM Signals . . . . .	70
4.5. OLT Subträgerbelegungsplan . . . . .	72
4.6. Zeitlicher Frameaufbau . . . . .	74
4.7. Gemeinsamer und Differentieller Phasenfehler . . . . .	76

4.8. Auswertung der Uplink Synchronisationssequenz am OLT	77
4.9. Aufbau des Kontrolldatenbereichs . . . . .	78
4.10. Nutzdaten Konfigurationsoptionen in Zeit- und Frequenzbereich . . . . .	80
4.11. Aufbau des Nutzdatenbereichs . . . . .	81
4.12. OLT Transmitter Architektur . . . . .	85
4.13. Unterstützte Modulationsformate . . . . .	85
4.14. ONU Empfänger Architektur . . . . .	87
4.15. ONU Synchronisationsmodul . . . . .	88
4.16. ONU Transmitter Architektur . . . . .	90
4.17. OLT Empfänger Architektur . . . . .	92
4.18. MAC Layer . . . . .	93
4.19. FPGA Gesamtsystem . . . . .	99
5.1. Struktureller Aufbau eines Radix-4 Algorithmus . . . . .	108
5.2. Konstantenmultiplikation auf Basis von Addition und Subtraktion . . . . .	111
5.3. DIT Radix-2 8 Punkt FFT . . . . .	113
5.4. DIF Radix-2 8 Punkt FFT . . . . .	114
5.5. 256 Punkt FFT in paralleler Ausführung . . . . .	115
5.6. 256 Punkt FFT in Multiplexausführung . . . . .	116
5.7. QAM-16 Konstellationsbereiche . . . . .	124
5.8. FPGA Interface für die Hochgeschwindigkeitswandler . . . . .	128
5.9. Logische Anbindung der ONU ADC Wandlerchips . . . . .	132
5.10. Logische Anbindung der ONU DAC Wandlerchips . . . . .	132
5.11. Paketformat der Inter-Kommunikationsschnittstelle . . . . .	135
5.12. Blockdiagramm der Inter-Kommunikationsschnittstelle . . . . .	136
6.1. Blockdiagramm des FPGA Mainboards . . . . .	142
6.2. Layout des FPGA Mainboards . . . . .	143
6.3. FPGA Mainboard . . . . .	144
6.4. Mainboard Spannungsversorgung . . . . .	147



6.5. Blockdiagramm eines ADC Moduls einer ONU . . . . .	149
6.6. Blockdiagramm eines DAC Moduls einer ONU . . . . .	150
6.7. ADC und DAC Module einer ONU . . . . .	151
6.8. Layout des DDS Taktgenerators . . . . .	153
6.9. Fertig aufgebauter DDS Taktgenerator . . . . .	153
6.10. Blockdiagramm der analogen Mischstufen der ONUs . . .	154
6.11. Analoge Mischstufen der ONUs . . . . .	157
6.12. Analoger Tiefpassfilter am ONU . . . . .	158
7.1. FFT Berechnungsgenauigkeit ohne Eingangsquantisie- rungsfehler . . . . .	162
7.2. FFT Berechnungsgenauigkeit mit Eingangsquantisie- rungsfehler . . . . .	163
7.3. SFP+ Interface Adapter . . . . .	166
7.4. Testaufbau zur Entwicklung der Wandlerinterfaces . . . .	168
7.5. Simulationsstrecke des Downlinks . . . . .	169
7.6. Ausgabe eines simulierten Downlinksignals durch einen AWG . . . . .	172
7.7. Test der Ethernetanbindung . . . . .	173
7.8. Kontrollfenster einer ONU . . . . .	174
7.9. Korrelations- und Energiesignal während dem Empfang einer ungestörten Synchronisationssequenz . . . . .	175
7.10. Differentieller Phasenfehler bei einem Frequenzoffset des Sampleoszillators . . . . .	176
7.11. Ausregelung eines gemeinsamen Phasenfehlers . . . . .	176



# Tabellenverzeichnis

2.1. Ressourcenübersicht der verwendete Virtex-6 FPGAs . . . . .	38
4.1. OFDM Systemparameter . . . . .	68
4.2. Aktive lokale Subträger während einer Phasenreferenz im Downlink . . . . .	73
4.3. Vollständige OFDM Systemparameter . . . . .	82
4.4. Overhead der OFDMA Prozessierung . . . . .	82
4.5. Overhead des Kommunikationsprotokolls . . . . .	83
4.6. Nutzdatenraten des Gesamtnetzwerks . . . . .	83
4.7. Nutzdatenraten pro spektraler Gruppe . . . . .	83
5.1. Verwendete Signaltypen . . . . .	104
5.2. Syntheseergebnisse der Radix-4 Butterfly Struktur . . . . .	108
5.3. Syntheseergebnisse der Konstantenmultiplikation . . . . .	112
5.4. Syntheseergebnisse der 256 Punkte FFT . . . . .	114
6.1. Mainboard Spannungsversorgung . . . . .	156
7.1. Gewählte Bitbreitenkonfiguration der 256 Punkte FFT . . . . .	161
7.2. FFT 256 Synthese Ergebnisse . . . . .	164
7.3. FFT 256 Place & Route Ergebnisse . . . . .	164



# Abkürzungsverzeichnis

<b>AD</b>	Analog-to-Digital
<b>ADC</b>	Analog-to-Digital-Converter
<b>AKF</b>	Autokorrelationsfunktion
<b>AGC</b>	Automatic Gain Control
<b>AM</b>	Amplitude-Modulation
<b>ASIC</b>	Application-Specific Integrated Circuit
<b>ASK</b>	Amplitude-Shift-Keying
<b>AWG</b>	Arbitrary Waveform Generator
<b>AHB</b>	Advanced High-performance Bus
<b>BPSK</b>	Binary-Phase-Shift-Keying
<b>CLB</b>	Configurable Logic Block
<b>CORDIC</b>	Coordinate Rotation Digital Computer
<b>CP</b>	Cyclic Prefix
<b>CPE</b>	Common Phase Error
<b>CPLD</b>	Complex Programmable Logic Device
<b>CPRI</b>	Common Public Radio Interface
<b>CPU</b>	Central Processing Unit
<b>CRC</b>	Cyclic Redundancy Check
<b>DC</b>	Direct Current
<b>DDR</b>	Double Data Rate
<b>DDR3</b>	Double Data Rate 3
<b>DDS</b>	Direct Digital Synthesis

<b>DA</b>	Digital-to-Analog
<b>DAC</b>	Digital-to-Analog Converter
<b>DPE</b>	Differential Phase Error
<b>DSL</b>	Digital Subscriber Line
<b>DSP</b>	Digital Signal Processing
<b>DFT</b>	Diskreten Fourier-Transformation
<b>DIT</b>	Decimation in Time
<b>DPSK</b>	Differential-Phase-Shift-Keying
<b>DIF</b>	Decimation in Frequency
<b>DVB-C</b>	Digital Video Broadcasting - Cable
<b>DVB-T</b>	Digital Video Broadcasting - Terrestrial
<b>EAM</b>	Electro-Absorption Modulator
<b>EV</b>	Error-Vector
<b>FDM</b>	Frequency-Division-Multiplexing
<b>FEC</b>	Forward Error Correction
<b>FFT</b>	Fast Fourier Transform
<b>FIFO</b>	First-In First-Out Memory
<b>FIR</b>	Finite Impulse Response
<b>FM</b>	Frequency-Modulation
<b>FMC</b>	FPGA Mezzanine Card
<b>FPGA</b>	Field Programmable Gate Array
<b>FPU</b>	Floating Point Unit
<b>FSK</b>	Frequency-Shift-Keying
<b>FSM</b>	Finite State Machine
<b>FTTH</b>	Fiber-To-The-Home
<b>FTTB</b>	Fiber-To-The-Building
<b>GPON</b>	Gigabit Passive Optical Network
<b>GPU</b>	Graphics Processing Unit

<b>GSM</b>	Global System for Mobile Communications
<b>IC</b>	Integrated Circuit
<b>IO</b>	Input/Output
<b>IQ</b>	in-phase and quadrature component
<b>IP</b>	Internet Protocol
<b>IFFT</b>	Inverse Fast Fourier Transform
<b>LDPC</b>	Low-Density Parity-Check
<b>LTE</b>	Long Term Evolution
<b>LVDS</b>	Low Voltage Differential Signaling
<b>MAC</b>	Medium Access Control
<b>MGL</b>	Multi-Gigabit-Link
<b>MGT</b>	Multi-Gigabit-Transceiver
<b>MIG</b>	Memory Interface Generator
<b>MIMO</b>	Multiple Input Multiple Output
<b>NCO</b>	Numerically Controlled Oscillator
<b>OFDM</b>	Orthogonal-Frequency-Division-Multiplexing
<b>OFDMA</b>	Orthogonal-Frequency-Division-Multiple-Access
<b>OLT</b>	Optical Line Termination
<b>ONU</b>	Optical Network Unit
<b>OOB</b>	Out-Of-Band
<b>OOK</b>	On-Off-Keying
<b>P2MP</b>	Punkt-zu-Mehrpunkt
<b>PAL</b>	Phase-Alternating-Line
<b>PAPR</b>	Peak-to-Average-Power-Ratio
<b>PC</b>	Personal Computer
<b>PLL</b>	Phase Locked Loop
<b>PM</b>	Phase-Modulation
<b>PSK</b>	Phase-Shift-Keying

## Abkürzungsverzeichnis

---

<b>PRBS</b>	Pseudorandom Binary Sequence
<b>PID</b>	Proportional-Integral-Derivative
<b>QoS</b>	Quality-of-Service
<b>QPSK</b>	Quadrature-Phase-Shift-Keying
<b>QAM</b>	Quadrature-Amplitude-Modulation
<b>ROM</b>	Read Only Memory
<b>SERDES</b>	Serializer/Deserializer
<b>SFP+</b>	Small Form-factor Pluggable Plus
<b>SC-FDMA</b>	Single-Carrier FDMA
<b>SoC</b>	System-on-Chip
<b>SO-DIMM</b>	Small Outline Dual Inline Memory Module
<b>SMA</b>	Sub-Miniature-A
<b>SNR</b>	Signal-to-Noise-Ratio
<b>SDR</b>	Software Defined Radio
<b>SPI</b>	Serial Peripheral Interface
<b>TCP</b>	Transmission Control Protocol
<b>TDMA</b>	Time Division Multiple Access
<b>TV</b>	Television
<b>UART</b>	Universal Asynchronous Receiver Transmitter
<b>VCO</b>	Voltage-Controlled-Oscillator
<b>VHDL</b>	Very High Speed Description Language
<b>VoIP</b>	Voice over IP
<b>WLAN</b>	Wireless-Local-Area-Network
<b>XGMII</b>	10 Gigabit Media Independent Interface



# Literatur- und Quellennachweise

- [1] *Hitech Global Xilinx Virtex 6 HXT 40G 100G Development Platform*, preliminary Aufl.
- [2] ABDULLAH, S., H. NAM, M. MCDERMOT und J. ABRAHAM: *A high throughput FFT processor with no multipliers*. In: *IEEE International Conference on Computer Design, 2009. ICCD 2009.*, S. 485–490, 2009.
- [3] ARM: *AMBA Specification (Rev 2.0)*, 1999.
- [4] ARMSTRONG, J.: *OFDM for Optical Communications*. *Lightwave Technology, Journal of*, 27(3):189–204, Feb 2009.
- [5] BERGLAND, G.: *Fast Fourier transform hardware implementations—An overview*. *IEEE Transactions on Audio and Electroacoustics*, 17(2):104–108, Juni 1969.
- [6] CHANG, R. W.: *Orthogonal frequency multiplex data transmission system*, 1970.
- [7] COLERI, S., M. ERGEN, A. PURI und A. BAHAI: *Channel estimation techniques based on pilot arrangement in OFDM systems*. *Broadcasting, IEEE Transactions on*, 48(3):223–229, Sep 2002.
- [8] COOLEY, J. und J. TUKEY: *An Algorithm for the Machine Calculation of Complex Fourier Series*. *Mathematics of Computation*, 19(90):297–301, 1965.
- [9] DESPAIN, A.: *Fourier Transform Computers Using CORDIC Iterations*. *IEEE Transactions on Computers*, C-23(10):993–1001, 1974.
- [10] DESPAIN, A.: *Very Fast Fourier Transform Algorithms Hardware for Implementation*. *IEEE Transactions on Computers*, C-28(5):333–341, Mai 1979.
- [11] DSP-TRICKS: *Computing inverse FFTs using the forward FFT*.
- [12] FILIPPI, A. und S. SERBETLI: *OFDM Symbol Synchronization Using*

- Frequency Domain Pilots in Time Domain*. In: *Vehicular Technology Conference, 2007. VTC-2007 Fall*. 2007 IEEE 66th, S. 1376–1380, Sept 2007.
- [13] GAISLER: *GRLIB IP Core User's Manual*. Aeroflex Gaisler, 1.3.7 Aufl., April 2014.
- [14] GAISLER: *GRLIB IP Library User's Manual*. Aeroflex Gaisler, 1.3.7 Aufl., April 2014.
- [15] GAISLER: *LEON/GRLIB Configuration and Development Guide*. Aeroflex Gaisler, April 2014.
- [16] GESBERT, D. und J. AKHTAR: *Breaking the barriers of Shannon's capacity - An overview of MIMO wireless systems*. *Signal Processing*, 1(B2):B3, 2002.
- [17] GIDDINGS, R. und J. TANG: *World-first experimental demonstration of synchronous clock recovery in an 11.25Gb/s real-time end-to-end optical OFDM system using directly modulated DFBs*. In: *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference*, S. 1–3, march 2011.
- [18] GIDDINGS, R. P., X. Q. JIN, E. HUGUES-SALAS, E. GIACOUMIDIS, J. L. WEI und J. M. TANG: *Experimental demonstration of a record high 11.25Gb/s real-time optical OFDM transceiver supporting 25km SMF end-to-end transmission in simple IMDD systems*. Bd. 18, S. 5541–5555. OSA, Mar 2010.
- [19] HILLERKUSS, D., M. WINTER, M. TESCHKE, A. MARCULESCU, J. LI, G. SIGURDSSON, K. WORMS, S. B. EZRA, N. NARKISS, W. FREUDE und J. LEUTHOLD: *Simple all-optical FFT scheme enabling Tbit/s real-time signal processing*. *Opt. Express*, 18(9):9324–9340, Apr 2010.
- [20] HUNG, C.-P. und Y. T. SU: *Joint Frequency and Symbol Synchronization Schemes for an OFDM System*. *Wireless Personal Communications* 10, 1999.
- [21] KANEDA, N., Q. YANG, X. LIU, W. SHIEH und Y.-K. CHEN: *Realizing real-time implementation of coherent optical OFDM receiver with FPGAs*. In: *Optical Communication, 2009. ECOC '09. 35th European Conference on*, S. 1–4, sept. 2009.
- [22] KAZOVSKY, L. G., W.-T. SHAW, D. GUTIERREZ, N. CHENG und

- S.-W. WONG: *Next-Generation Optical Access Networks*. *Lightwave Technology, Journal of*, 25(11):3428–3442, Nov 2007.
- [23] LI, X. und L. CIMINI: *Effects of clipping and filtering on the performance of OFDM*. In: *Vehicular Technology Conference, 1997, IEEE 47th*, Bd. 3, S. 1634–1638 vol.3, May 1997.
- [24] MEYER, J. H.: *HW/SW Co-Design Framework für Hochgeschwindigkeits-OFDM Signalverarbeitung*. Dissertation, Karlsruhe Institute of Technology, 2014.
- [25] MICRAM: *VEGA ADC30 Datasheet*, preliminary Aufl.
- [26] MICRAM: *VEGA DAC30 Datasheet*, preliminary Aufl.
- [27] NEBENDAHL, B., W. FREUDE, C. KOOS, J. LEUTHOLD, M. HUEBNER, R. SCHMOGROW, A. JOSTEN, D. HILLERKUSS, S. KOENIG, M. WINTER, W. FREUDE, C. KOOS, J. LEUTHOLD, J. MEYER, M. DRESCHMANN, M. HUEBNER, D. HILLERKUSS, J. LEUTHOLD und M. WINTER: *EVM as new quality metric for optical modulation analysis*. In: *Electronics, Communications and Photonics Conference (SIECPC), 2013 Saudi International*, S. 1–4, April 2013.
- [28] NETO, L., P. CHANLOU, B. CHARBONNIER, N. GENAY, F. SALIOU, R. XIA, M. OUZZIF, C. AUPETIT-BERTHELEMOT, J. LE MASSON, E. GRARD und V. RODRIGUES: *Up to 40Gb/s optically amplified AMOOFDM for next generation PON networks*. In: *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference*, S. 1–3, March 2011.
- [29] ONOHARA, K., T. SUGIHARA, Y. MIYATA, K. SUGIHARA, K. KUBO, H. YOSHIDA, K. KOGUCHI und T. MIZUOCHI: *Soft-decision forward error correction for 100 Gb/s digital coherent systems*. *Optical Fiber Technology*, 17(5):452 – 455, 2011. 100G and Beyond.
- [30] OPEN-CORES: *Ethernet 10GE MAC*.
- [31] OPEN-CORES: *Reed-Solomon Codec Generator*.
- [32] PENG, X. und C. J. SHU: *FPGA Implementation of High Speed FFT Algorithm Based on Split-Radix*. In: *International Symposium on Intelligent Information Technology Application Workshops, 2008. IITAW '08.*, S. 781–784, 2008.
- [33] PETROVSKY, A. und S. SHKREDOV: *Automatic Generation of Split-*

- Radix 2-4 Parallel-Pipeline FFT Processors: Hardware Reconfiguration and Core Optimizations.* In: *International Symposium on Parallel Computing in Electrical Engineering*, 2006. PAR ELEC 2006., S. 181–186, 2006.
- [34] QURESHI, F. und O. GUSTAFSSON: *Low-complexity reconfigurable complex constant multiplication for FFTs.* In: *IEEE International Symposium on Circuits and Systems*, 2009. ISCAS 2009., S. 1137–1140, Mai 2009.
- [35] SCHMIDL, T. und D. COX: *Robust frequency and timing synchronization for OFDM.* *Communications*, IEEE Transactions on, 45(12):1613–1621, 1997.
- [36] SCHMOGROW, R., M. WINTER, D. HILLERKUSS, B. NEBENDAHL, S. BEN EZRA, J. MEYER, M. DRESCHMANN, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Real-time OFDM transmitter beyond 100 Gbit/s.* *Opt. Express*, 19(13):12740–12749, Jun 2011.
- [37] SCHMOGROW, R., M. WINTER, D. HILLERKUSS, B. NEBENDAHL, S. BEN-EZRA, J. MEYER, M. DRESCHMANN, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Real-time OFDM transmitter beyond 100 Gbit/s.* In: *Opt. Express* 19(13), pp. 12740-12749, 2011.
- [38] SHANNON, C. E.: *A Mathematical Theory of Communication.* *The Bell System Technical Journal*, 27(3):379–423, 1948.
- [39] SMITH, D.: *Reducing the optical component cost for future fibre access.* In: *Optical Communication*, 2009. ECOC '09. 35th European Conference on, S. 1–2, Sept 2009.
- [40] TRANSMODE: *A Transmode iWDM-PON access network.*
- [41] WEINSTEIN, S.: *The history of orthogonal frequency-division multiplexing [History of Communications].* *Communications Magazine*, IEEE, 47(11):26–35, November 2009.
- [42] WEINSTEIN, S. und P. EBERT: *Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform.* *Communication Technology*, IEEE Transactions on, 19(5):628–634, October 1971.
- [43] WIKIPEDIA: *Serial Peripheral Interface.*
- [44] XILINX: *Virtex-6 FPGA GTX Transceivers User Guide (UG366).* Xilinx, 2.6 Aufl., July 2011.

- [45] XILINX: *Virtex-6 FPGA Memory Interface Solutions User Guide (UG406)*. Xilinx, March 2013.
- [46] YEH, H.-G. und G. TRUONG: *Speed and area analysis of memory based FFT processors in a FPGA*. In: *Wireless Telecommunications Symposium, 2007. WTS 2007*, S. 1 –6, 2007.
- [47] ZHANG, G. und F. CHEN: *Parallel FFT with CORDIC for ultra wide band*. In: *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004.*, Bd. 2, S. 1173 – 1177 Vol.2, 2004.
- [48] ZHOU, E., X. ZHANG, H. ZHAO und W. WANG: *Synchronization algorithms for MIMO OFDM systems*. In: *Wireless Communications and Networking Conference, 2005 IEEE*, Bd. 1, S. 18–22 Vol. 1, March 2005.



# Betreute studentische Arbeiten

- [Abr14] ABRECHT, FELIX: *Carrier Recovery for Optical Real-Time Receivers*. Masterarbeit, Karlsruher Institut für Technologie, Institut für Technik der Informationsverarbeitung, 2014.
- [Bla15] BLASE, BENJAMIN: *Entwicklung eines FPGA-basierten Time-to-Digital-Converters*. Masterarbeit, Karlsruher Institut für Technologie, Institut für Technik der Informationsverarbeitung, 2015.
- [Hel12] HELD, FELIX: *Digitale Audio-, Video- & Licht-Kommunikationsprotokolle in der Veranstaltungstechnik*. Seminararbeit, Karlsruher Institut für Technologie, Institut für Technik der Informationsverarbeitung, 2012.
- [Luc14] LUCIANO, MARIO MARQUEZ: *Konzept und Implementierung einer FPGA basierten Traktionskontrolle und Torque Vectoring für den Einsatz in Elektrofahrzeugen*. Masterarbeit, Karlsruher Institut für Technologie, Institut für Technik der Informationsverarbeitung, 2014.
- [Ruh11] RUHLAND, CHRISTOPH: *Design und Implementierung eines FPGA-basierten Highspeed OFDM Transmitters*. Diplomarbeit ID-1487, Karlsruher Institut für Technologie, Institut für Technik der Informationsverarbeitung, 2011.
- [Ste12] STEIN, CARSTEN: *Verwendung multipler GBit Ethernet Links zur digitalen Übermittlung von Analogsignalen*. Studienarbeit, Karlsruher Institut für Technologie, Institut für Technik der Informationsverarbeitung, 2012.
- [Tra10] TRADOWSKY, CARSTEN: *Design und Implementierung eines Testprozessors auf Basis der Xilinx Picoblaze Architektur*. Studienarbeit, Karlsruher Institut für Technologie, Institut für Technik der Informationsverarbeitung, 2010.





# Eigene Veröffentlichungen

## Konferenzbeiträge, Journalbeiträge und Buchkapitel

- [BDK<sup>+</sup>08] BOETTGER, G., M. DRESCHMANN, C. KLAMOURIS, M. HUEBNER, M. ROEGER, A.W. BETT, T. KUENG, J. BECKER, W. FREUDE und J. LEUTHOLD: *An Optically Powered Video Camera Link*. *Photonics Technology Letters, IEEE*, 20(1):39–41, Jan 2008.
- [BDR<sup>+</sup>07] BOETTGER, G., M. DRESCHMANN, M. ROEGER, M. HUEBNER, C. KLAMOURIS, A. W. BETT, T. KUENG, J. BECKER, W. FREUDE und J. LEUTHOLD: *Optically Powered Video Camera Link*. In: *Optical Communication (ECOC), 2007 33rd European Conference and Exhibition of*, Seiten 1–2, Sept 2007.
- [BHD<sup>+</sup>06] BOETTGER, G., M. HUEBNER, M. DRESCHMANN, C. KLAMOURIS, K. PAULSSON, T. KUENG, A. W. BETT, J. BECKER, W. FREUDE und J. LEUTHOLD: *Optically Powered Video Camera Network*. 13. VDE-ITG-Fachtagung Kommunikationskabelnetze. 12.–13. Dez. 2006 ITG-Fachbericht Band 197, Seiten pp.123–124, 2006.
- [BJA<sup>+</sup>15] BAEUERLE, B., A. JOSTEN, F. ABRECHT, E. DORNBIERER, J. BOESSER, M. DRESCHMANN, J. BECKER, J. LEUTHOLD und D. HILLERKUSS: *Multiplier-free carrier-phase recovery for real-time receivers using processing in polar coordinates*. In: *Optical Fiber Communications Conference and Exhibition (OFC), 2015*, Seiten 1–3, March 2015.
- [CKD<sup>+</sup>09] CAMPI, F., R. KOENIG, M. DRESCHMANN, M. NEUKIRCH-

- NER, D. PICARD, M. JUETTNER, E. SCHULER, A. DELEDDA, D. ROSSI, A. PASINI, M. HUEBNER, J. BECKER und R. GUERRIERI: *RTL-to-layout implementation of an embedded coarse grained architecture for dynamically reconfigurable computing in systems-on-chip*. In: *System-on-Chip, 2009. SOC 2009. International Symposium on*, Seiten 110–113, Oct 2009.
- [DHG<sup>+</sup>15] DRESCHMANN, M., J. HEISSWOLF, M. GEIGER, M. HAUSSECKER und J. BECKER: *A Framework for Multi-FPGA Interconnection using Multi Gigabit Transceivers*. In: *Integrated Circuits and Systems Design (SBCCI), 2015 28th Symposium on*, August 2015.
- [DHR<sup>+</sup>10] DRESCHMANN, M., M. HUEBNER, M. ROEGER, O. SANDER, C. KLAMOURIS, J. BECKER, W. FREUDE und J. LEUTHOLD: *Reconfigurable Hardware for Power-over-Fiber Applications*. In: *Field Programmable Logic and Applications (FPL), 2010 International Conference on*, Seiten 525–531, Aug 2010.
- [DMH<sup>+</sup>11] DRESCHMANN, M., J. MEYER, M. HUEBNER, R. SCHMOGROW, D. HILLERKUSS, J. BECKER, J. LEUTHOLD und W. FREUDE: *Implementation of an ultra-high speed 256-point FFT for Xilinx Virtex-6 devices*. In: *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, Seiten 829–834, July 2011.
- [DMH<sup>+</sup>12] DRESCHMANN, M., J. MEYER, M. HUBNER, R. SCHMOGROW, D. HILLERKUSS, J. BECKER, J. LEUTHOLD und W. FREUDE: *Time and frequency synchronization for ultra-high speed OFDM systems*. In: *Computing, Networking and Communications (ICNC), 2012 International Conference on*, Seiten 871–875, Jan 2012.
- [DMS<sup>+</sup>14] DRESCHMANN, M., J. MEYER, P. SCHINDLER, R. SCHMOGROW, J. LEUTHOLD, W. FREUDE und J. BECKER: *An ultra-high speed OFDMA system for optical access networks*. In: *Computing, Networking and Communications (ICNC), 2014 International Conference on*, Seiten 889–894, Feb 2014.
- [DSK<sup>+</sup>13] DRESCHMANN, M., O. SANDER, A. KLIMM, C. ROTH und J. BECKER: *Addiguration: Exploring configuration behavior of*

*Spartan-3 devices*. In: *Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), 2013 8th International Workshop on*, Seiten 1–6, July 2013.

- [FHS<sup>+</sup>10] FREUDE, W., D. HILLERKUSS, T. SCHELLINGER, R. SCHMOGROW, M. WINTER, T. VALLAITIS, R. BONK, A. MARCULESCU, M. DRESCHMANN, J. MEYER, S. BEN EZRA, N. NARKISS, B. NEBENDAHL, F. PARMIGIANI, P. PETROPOULOS, B. RESAN, K. WEINGARTEN, T. ELLERMAYER, J. LUTZ, M. MOELLER, M. HUEBNER, J. BECKER, C. KOOS und J. LEUTHOLD: *Orthogonal frequency division multiplexing (OFDM) in photonic communications*. In: *The International Conference on Fiber Optics and Photonics (Photonics 2010)*, Dezember 2010.
- [FHS<sup>+</sup>11] FREUDE, W., D. HILLERKUSS, T. SCHELLINGER, R. SCHMOGROW, M. WINTER, T. VALLAITIS, R. BONK, A. MARCULESCU, J. LI, M. DRESCHMANN, J. MEYER, S. BEN EZRA, M. CASPI, B. NEBENDAHL, F. PARMIGIANI, P. PETROPOIU, B. RESAN, A. OEHLER, K. WEINGARTEN, T. ELLERMAYER, J. LUTZ, M. MOLLER, M. HUEBNER, J. BECKER, C. KOOS und J. LEUTHOLD: *All-optical real-time OFDM transmitter and receiver*. In: *Lasers and Electro-Optics (CLEO), 2011 Conference on*, Seiten 1–2, May 2011.
- [FRD<sup>+</sup>09] FREUDE, W., M. ROEGER, M. DRESCHMANN, M. HUEBNER, A.W. BETT, J. BECKER und J. LEUTHOLD: *An optically powered fibre network for heterogeneous subscribers*. In: *Transparent Optical Networks, 2009. ICTON '09. 11th International Conference on*, Seiten 1–4, June 2009.
- [FSH<sup>+</sup>12] FREUDE, W., R. SCHMOGROW, D. HILLERKUSS, J. MEYER, M. DRESCHMANN, B. NEBENDAHL, M. HUEBNER, J. BECKER, C. KOOS und J. LEUTHOLD: *Reconfigurable optical transmitters and receivers*. SPIE Proceedings, "8284":82840A–82840A–8, 2012.
- [FSJ<sup>+</sup>12] FREUDE, W., R. SCHMOGROW, A. JOSTEN, D. HILLERKUSS, S. KOENIG, J. MEYER, M. DRESCHMANN, C. KOOS, J. LEUTHOLD, B. NEBENDAHL, M. WINTER und M. HUEBNER: *Quality metrics for optical transmission*. In: *Photonics in Switching*

(PS), 2012 International Conference on, Seiten 1–4, Sept 2012.

- [FSN<sup>+</sup>11] FREUDE, W., R. SCHMOGROW, B. NEBENDAHL, D. HILLERKUSS, J. MEYER, M. DRESCHMANN, M. HUEBNER, J. BECKER, C. KOOS und J. LEUTHOLD: *Software-defined optical transmission*. In: *Transparent Optical Networks (ICTON), 2011 13th International Conference on*, Seiten 1–4, June 2011.
- [FSN<sup>+</sup>12] FREUDE, W., R. SCHMOGROW, B. NEBENDAHL, M. WINTER, A. JOSTEN, D. HILLERKUSS, S. KOENIG, J. MEYER, M. DRESCHMANN, M. HUEBNER, C. KOOS, J. BECKER und J. LEUTHOLD: *Quality metrics for optical signals: Eye diagram, Q-factor, OSNR, EVM and BER*. In: *Transparent Optical Networks (ICTON), 2012 14th International Conference on*, Seiten 1–4, July 2012.
- [HSM<sup>+</sup>12] HILLERKUSS, D., R. SCHMOGROW, M. MEYER, S. WOLF, M. JORDAN, P. KLEINOW, N. LINDENMANN, P.C. SCHINDLER, A. MELIKYAN, XIN YANG, S. BEN-EZRA, B. NEBENDAHL, M. DRESCHMANN, J. MEYER, F. PARMIGIANI, P. PETROPOULOS, B. RESAN, A. OEHLER, K. WEINGARTEN, L. ALTENHAIN, T. ELLERMAYER, M. MOELLER, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Single-laser 32.5 Tbit/s Nyquist WDM transmission*. *Optical Communications and Networking, IEEE/OSA Journal of*, 4(10):715–723, Oct 2012.
- [HSS<sup>+</sup>10] HILLERKUSS, D., T. SCHELLINGER, R. SCHMOGROW, M. WINTER, T. VALLAITIS, R. BONK, A. MARCULESCU, J. LI, M. DRESCHMANN, J. MEYER, S. BEN EZRA, N. NARKISS, B. NEBENDAHL, F. PARMIGIANI, P. PETROPOULOS, B. RESAN, K. WEINGARTEN, T. ELLERMAYER, J. LUTZ, M. MOLLER, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Single source optical OFDM transmitter and optical FFT receiver demonstrated at line rates of 5.4 and 10.8 Tbit/s*. In: *Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC)*, Seiten 1–3, March 2010.
- [HSS<sup>+</sup>11] HILLERKUSS, D., R. SCHMOGROW, T. SCHELLINGER, M. JORDAN, M. WINTER, G. HUBER, T. VALLAITIS, R. BONK,

- F. KLEINOW, F. FREY, M. ROEGER, S. KOENIG, A. LUDWIG, A. MARCULESCU, J. LI, M. HOH, M. DRESCHMANN, J. MEYER, S. BEN EZRA, N. NARKISS, B. NEBENDAHL, F. PARMIGIANI, P. PETROPOULOS, B. RESAN, A. OEHLER, K. WEINGARTEN, T. ELLERMEYER, J. LUTZ, M. MOELLER, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *26 Tbit/s line-rate super-channel transmission utilizing all-optical fast Fourier transform processing*. *Nature Photonics*, 5(6):364–371, June 2011.
- [KBH<sup>+</sup>06] KLAMOURIS, C., G. BOETTGER, M. HUEBNER, M. DRESCHMANN, K. PAULSSON, A.W. BETT, T. KUENG, J. BECKER, W. FREUDE und J. LEUTHOLD: *Optically Powered Platform with Mb/s Transmission over a Single Fiber*. In: *Optical Communications, 2006. ECOC 2006. European Conference on*, Seiten 1–2, Sept 2006.
- [KTK<sup>+</sup>13] KANONAKIS, K., I. TOMKOS, H.-G. KRIMMEL, F. SCHAICH, C. LANGE, E. WEIS, M. DRESCHMANN, R. SCHMOGROW, P. KOURTESSIS, M. MILOSAVLJEVIC, I. CANO, J. PRAT und J.A. TORRIJOS GIJON: *Results from the EU project ACCORDANCE on converged OFDMA-PON networks*. In: *Transparent Optical Networks (ICTON), 2013 15th International Conference on*, Seiten 1–4, June 2013.
- [LWF<sup>+</sup>10] LEUTHOLD, J., M. WINTER, W. FREUDE, C. KOOS, D. HILLERKUSS, T. SCHELLINGER, R. SCHMOGROW, T. VALLAITIS, R. BONK, A. MARCULESCU, J. LI, M. DRESCHMANN, J. MEYER, M. HUEBNER, J. BECKER, S. BEN EZRA, N. NARKISS, B. NEBENDAHL, F. PARMIGIANI, P. PETROPOULOS, B. RESAN, A. OEHLER, K. WEINGARTEN, T. ELLERMEYER, J. LUTZ und M. MOELLER: *All-Optical FTT Signal Processing of a 10.8 Tb/s Single Channel OFDM Signal*. In: *Photonics in Switching*, Seite PWC1. Optical Society of America, Juli 2010.
- [MDK<sup>+</sup>13] MEYER, J., M. DRESCHMANN, D. KARNICK, P.C. SCHINDLER, W. FREUDE, J. LEUTHOLD und J. BECKER: *A novel system on chip for software-defined, high-speed OFDM signal processing*. In: *Integrated Circuits and Systems Design (SBCCI), 2013 26th Symposium on*, Seiten 1–6, Sept 2013.

- [MMD<sup>+</sup>12] MEYER, JOACHIM, SIMON MENZEL, MICHAEL DRESCHMANN, RENE SCHMOGROW, DAVID HILLERKUSS, WOLFGANG FREUDE, JUERG LEUTHOLD und JUERGEN BECKER: *Ultra High Speed Digital Down Converter Design for Virtex-6 FPGAs*. In: *OFDM 2012, 17th International OFDM Workshop 2012 (InOWo'12); Proceedings of*, Seiten 1–5, Aug 2012.
- [MSA<sup>+</sup>14] MEDER, L., P.C. SCHINDLER, A. AGMON, M. MELTSIN, R. BONK, J. MEYER, M. DRESCHMANN, A. TOLMACHEV, R. HILGENDORF, M. NAZARATHY, S. BEN-EZRA, T. PFEIFFER, W. FREUDE, J. LEUTHOLD, C. KOOS und J. BECKER: *Flexible real-time transmitter at 10 Gbit/s for SCFDMA PONs focusing on low-cost ONUs*. In: *Design and Architectures for Signal and Image Processing (DASIP), 2014 Conference on*, Seiten 1–8, Oct 2014.
- [NFK<sup>+</sup>13] NEBENDAHL, B., W. FREUDE, C. KOOS, J. LEUTHOLD, M. HUEBNER, R. SCHMOGROW, A. JOSTEN, D. HILLERKUSS, S. KOENIG, M. WINTER, W. FREUDE, C. KOOS, J. LEUTHOLD, J. MEYER, M. DRESCHMANN, M. HUEBNER, D. HILLERKUSS, J. LEUTHOLD und M. WINTER: *EVM as new quality metric for optical modulation analysis*. In: *Electronics, Communications and Photonics Conference (SIECPC), 2013 Saudi International*, Seiten 1–4, April 2013.
- [NSD<sup>+</sup>12] NEBENDAHL, B., R. SCHMOGROW, T. DENNIS, A. JOSTEN, D. HILLERKUSS, S. KOENIG, J. MEYER, M. DRESCHMANN, M. WINTER, M. HUEBNER, W. FREUDE, C. KOOS und J. LEUTHOLD: *Quality Metrics in optical modulation analysis: EVM and its relation to Q-factor, OSNR, and BER*. In: *Communications and Photonics Conference (ACP), 2012 Asia*, Seiten 1–3, Nov 2012.
- [PHA<sup>+</sup>07] PAULSSON, K., M. HUBNER, G. AUER, M. DRESCHMANN, L. CHEN und J. BECKER: *Implementation of a Virtual Internal Configuration Access Port (JCAP) for Enabling Partial Self-Reconfiguration on Xilinx Spartan III FPGAs*. In: *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, Seiten 351–356, Aug 2007.
- [RBD<sup>+</sup>08] ROEGER, M., G. BOETTGER, M. DRESCHMANN, C. KLAMOU-

- RIS, M. HUEBNER, A. W. BETT, J. BECKER, W. FREUDE und J. LEUTHOLD: *Optically powered fiber networks*. Optics Express, 16:21821–21834, Dezember 2008.
- [SAW<sup>+</sup>15] SCHINDLER, P.C., A. AGMON, S. WOLF, R. BONK, L. MEDER, M. MELTSIN, A. LUDWIG, R. SCHMOGROW, M. DRESCHMANN, J. MEYER, J. BECKER, M. NAZARATHY, S. BEN-EZRA, T. PFEIFFER, W. FREUDE, J. LEUTHOLD und C. KOOS: *Ultra-Dense, Single-Wavelength DFT-Spread OFDMA PON With Laserless 1.2 Gb/s ONU Ready for Silicon Photonics Integration*. Lightwave Technology, Journal of, 33(8):1650–1659, April 2015.
- [SBH<sup>+</sup>07] SANDER, O., J. BECKER, M. HUEBNER, M. DRESCHMANN, J. LUKA, M. TRAUB und T. WEBER: *Modular system concept for FPGA-based Automotive Gateway*. Electronic Systems for Vehicles, ser. VDI-Berichte, no. 2000, Verein Deutscher Ingenieure, pp. 223-232, 2007.
- [SHD<sup>+</sup>10] SCHMOGROW, R., D. HILLERKUSS, M. DRESCHMANN, M. HUEBNER, M. WINTER, J. MEYER, B. NEBENDAHL, C. KOOS, J. BECKER, W. FREUDE und J. LEUTHOLD: *Real-Time Software-Defined Multiformat Transmitter Generating 64QAM at 28 Gbd*. Photonics Technology Letters, IEEE, 22(21):1601–1603, Nov 2010.
- [SMS<sup>+</sup>14] SCHMOGROW, R., M. MEYER, P. C. SCHINDLER, B. NEBENDAHL, M. DRESCHMANN, J. MEYER, A. JOSTEN, D. HILLERKUSS, S. BEN EZRA, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Real-time Nyquist signaling with dynamic precision and flexible non-integer oversampling*. Opt. Express, 22(1):193–209, Jan 2014.
- [SMW<sup>+</sup>12] SCHMOGROW, R., M. MEYER, S. WOLF, B. NEBENDAHL, D. HILLERKUSS, B. BAEUERLE, M. DRESCHMANN, J. MEYER, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *150 Gbit/s real-time Nyquist pulse transmission Over 150 km SSMF enhanced by DSP with dynamic precision*. In: *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2012 and the National Fiber Optic Engineers Conference*, Seiten 1–3, March 2012.

- [SNW<sup>+</sup>12] SCHMOGROW, R., B. NEBENDAHL, M. WINTER, A. JOSTEN, D. HILLERKUSS, S. KOENIG, J. MEYER, M. DRESCHMANN, M. HUEBNER, C. KOOS, J. BECKER, W. FREUDE und J. LEUTHOLD: *Error Vector Magnitude as a Performance Measure for Advanced Modulation Formats*. *Photonics Technology Letters*, IEEE, 24(1):61–63, Jan 2012.
- [SSD<sup>+</sup>13a] SCHINDLER, P.C., R. SCHMOGROW, M. DRESCHMANN, J. MEYER, D. HILLERKUSS, I. TOMKOS, J. PRAT, H.-G. KRIMMEL, T. PFEIFFER, P. KOURTESSIS, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Flexible WDM-PON with Nyquist-FDM and 31.25 Gbit/s per wavelength channel using colorless, low-speed ONUs*. In: *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC), 2013*, Seiten 1–3, March 2013.
- [SSD<sup>+</sup>13b] SCHINDLER, P.C., R. SCHMOGROW, M. DRESCHMANN, J. MEYER, I. TOMKOS, J. PRAT, H.-G. KRIMMEL, T. PFEIFFER, P. KOURTESSIS, A. LUDWIG, D. KARNICK, D. HILLERKUSS, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Colorless FDMA-PON with flexible bandwidth allocation and colorless, low-speed ONUs [invited]*. *Optical Communications and Networking*, IEEE/OSA Journal of, 5(10):A204–A212, Oct 2013.
- [STD<sup>+</sup>08] SANDER, O., M. TRAUB, M. DRESCHMANN, M. HUEBNER, J. LUKA, T. WEBER und J. BECKER: *Modular system concept for FPGA-based Automotive Gateway*. Nummer 2000 in *VDI-Berichte*. Verein Deutscher Ingenieure, 2008.
- [SWH<sup>+</sup>11] SCHMOGROW, R., M. WINTER, D. HILLERKUSS, B. NEBENDAHL, S. BEN EZRA, J. MEYER, M. DRESCHMANN, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Real-time OFDM transmitter beyond 100 Gbit/s*. *Opt. Express*, 19(13):12740–12749, Jun 2011.
- [SWM<sup>+</sup>11] SCHMOGROW, R., M. WINTER, M. MEYER, D. HILLERKUSS, B. NEBENDAHL, J. MEYER, M. DRESCHMANN, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Real-Time Nyquist Pulse Modulation Transmitter Generating Rectangular Shaped Spectra of 112 Gbit/s 16QAM Signals*. In: *Signal Processing in Photonic Communications*, Seite SPMA5. *Optical*



Society of America, 2011.

- [SWM<sup>+</sup>12] SCHMOGROW, R., M. WINTER, M. MEYER, D. HILLERKUSS, S. WOLF, B. BAEUERLE, A. LUDWIG, B. NEBENDAHL, S. BEN EZRA, J. MEYER, M. DRESCHMANN, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *Real-time Nyquist pulse generation beyond 100 Gbit/s and its relation to OFDM*. Opt. Express, 20(1):317–337, Jan 2012.
- [SWN<sup>+</sup>11] SCHMOGROW, R., M. WINTER, B. NEBENDAHL, D. HILLERKUSS, J. MEYER, M. DRESCHMANN, M. HUEBNER, J. BECKER, C. KOOS, W. FREUDE und J. LEUTHOLD: *101.5 Gbit/s real-time OFDM transmitter with 16QAM modulated subcarriers*. In: *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference*, Seiten 1–3, March 2011.

## Patente

- [JPB09a] JÜRGEN PROF. BECKER, MICHAEL DRESCHMANN, MICHAEL DR.-ING. HÜBNER JÜRGEN DIPL.-ING. LUKA OLIVER DIPL.-ING. SANDER MATTHIAS DIPL.-ING. TRAUB THOMAS DIPL.-ING. WEBER: *Data exchange device i.e. communication structure, for e.g. application specific integrated circuit, has function modules for processing interface-related functions, and master unit including number of signal inputs*, 2009.
- [JPB09b] JÜRGEN PROF. BECKER, MICHAEL DRESCHMANN, MICHAEL DR.-ING. HÜBNER JÜRGEN DIPL.-ING. LUKA OLIVER DIPL.-ING. SANDER MATTHIAS DIPL.-ING. TRAUB THOMAS DIPL.-ING. WEBER: *Routing module i.e. hardware-based routing module, for exchanging data between e.g. controller area network bus systems, and/or functional systems of communication network in motor vehicle, has routing unit formed as integrated circuit*, 2009.