



Daniel Rother

**Optimierung einer Software-Defined-
Radio-Plattform am Beispiel des Systems
Tower Overlay over LTE-A+**



Technische
Universität
Braunschweig



Institut für Nachrichtentechnik

Optimierung einer Software-Defined-Radio-Plattform am Beispiel des Systems Tower Overlay over LTE-A+

Von der Fakultät für Elektrotechnik, Informationstechnik, Physik
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines Doktors
der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von Daniel Rüdiger Rother, M.Sc.

aus Marburg

eingereicht am: 25.04.2018

mündliche Prüfung am: 08.01.2019

1. Referent: Prof. Dr.-Ing. Ulrich Reimers
2. Referent: Prof. Dr-Ing. Rolf Ernst
3. Referent: Prof. Dr.-Ing. Thomas Kürner (Vorsitz)

Druckjahr: 2019

**Dissertation an der Technischen Universität Braunschweig,
Fakultät für Elektrotechnik, Informationstechnik, Physik**

Mitteilungen aus dem Institut für Nachrichtentechnik der
Technischen Universität Braunschweig

Band 60

Daniel Rother

**Optimierung einer Software-Defined-Radio-Plattform
am Beispiel des Systems Tower Overlay over LTE-A+**

Shaker Verlag
Düren 2019

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Braunschweig, Techn. Univ., Diss., 2019

Copyright Shaker Verlag 2019

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-6778-1

ISSN 1865-2484

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren
Telefon: 02421 / 99 0 11 - 0 • Telefax: 02421 / 99 0 11 - 9
Internet: www.shaker.de • E-Mail: info@shaker.de

Vorwort

Diese Dissertation entstand in meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Nachrichtentechnik (IfN) der TU Braunschweig. Eine Arbeit wie diese ist nicht denkbar ohne die Unterstützung von Kollegen, Freunden und Familie. Ich freue mich deshalb, all denen danken zu können, die zum erfolgreichen Gelingen dieser Dissertation beigetragen haben.

Zuallererst möchte ich mich ganz herzlich für die intensive fachliche und persönliche Betreuung bei meinem Doktorvater Prof. Dr-Ing. Ulrich Reimers bedanken. Er hat mich in vielen Gesprächen mit unzähligen Anregungen während aller Phasen der Promotion großartig unterstützt und damit einen wesentlichen Beitrag zum Gelingen dieser Arbeit geleistet. Darüber hinaus hat er als Institutsleiter für die außergewöhnlich gute Arbeitsatmosphäre am IfN gesorgt. Mein weiterer Dank gilt Prof. Rolf Ernst für die Übernahme des Zweitgutachtens der Dissertation und Prof. Thomas Kürner, der den Vorsitz der Prüfungskommission übernommen hat.

Ein Großteil der Forschungsergebnisse dieser Dissertation basieren auf Arbeiten, die ich im Rahmen des Tower Overlay (TOoL+) Projektes durchgeführt habe. Ich danke daher allen an diesem Projekt beteiligten Personen, insbesondere unseren Projektpartnern Télédiffusion de France (TDF), Radiotelevisione Italiana (Rai) und GatesAir, für die gemeinsamen Arbeiten und die Möglichkeit, TOoL+ unter realen Bedingungen erproben zu können.

Ein großer Dank gilt natürlich auch dem TOoL+ Team Stefan Ilsen, Frieder Juretzek und Lucca Richter, den (Zimmer-)Kollegen aus der Mediengruppe Mark Hoyer, Florian Jackisch, Fabian Schrieber, Simon Walz und Jan Zöllner sowie allen übrigen IfNlern. Vielen Dank für die erfolgreiche gemeinsame Arbeit, das Korrekturlesen und für die schöne gemeinsame Zeit am IfN. Mit letzterer verbinde ich nicht nur die Forschung, sondern auch viele abwechslungsreiche und schöne Momente, auch außerhalb der Arbeitszeit. Kollegen wurden zu Freunden und begleiten mich auch heute noch. Nicht vergessen werden dürfen auch die vielen „guten Seelen“ des IfNs, u.a. Nina Andersen, Bogusia Brandt, Andreas Gudat, Rudolf Görke, Ronja Haase und Peter Schlegel, die mit ihrer täglichen Arbeit den Institutsbetrieb am Laufen halten. Ich schätze mich glücklich, am IfN auf dem Weg zu meiner Promotion so viel Unterstützung erhalten zu haben. An alle, nochmals vielen Dank.

Bedanken möchte ich mich auch bei den zahlreichen Studierenden, die im Rahmen ihrer Bachelor- und Masterarbeiten oder als HiWi wertvolle Zuarbeiten sowohl für die Projektarbeit als auch für meine Dissertation geleistet haben.

Meinem bestem Freund Martin Stucki danke ich für die zahlreichen (fachlichen) Gespräche über die Promotion, die Motivationshilfen, das Korrekturlesen und für die Vermittlung an die Maschinenbauer(innen) der Saalgemeinschaft Kette der Uni Hannover. Diese haben mir zum Zusammenschreiben dieser Dissertation „Asyl“ gewährt; auch dafür vielen herzlichen Dank.

Meinen Eltern danke ich für den Rückhalt, den sie mir bereits während des Studiums immer gegeben haben. Abschließend sei gesagt, dass der wichtigste Mensch in der Zeit meiner Promotion ohne Zweifel meine Frau Sandra gewesen ist. Ich bin mir sicher, dass ich ohne sie und ihre Unterstützung die Promotion nicht geschafft hätte. Vielen Dank!

Hannover, im Mai 2019

Daniel Rother

Kurzfassung

Heutige zellulare Mobilfunknetze erreichen durch die steigende Nachfrage nach Videodiensten ihre Kapazitätsgrenzen. Statt einer Nutzer-individuellen Unicast-Übertragung, ist die Verwendung von Broadcast-/Multicast-Ansätzen und den aus der klassischen TV- und Radio-Rundfunkübertragung bekannten hohen Sendern mit großer Sendeleistung für diesen Anwendungsfall Ressourcen und Kosten effizienter. Um dies auf Basis existierender Mobilfunksysteme zu realisieren, wurde das so genannte *Tower Overlay over LTE-A+* (TOoL+) entwickelt und mit Hilfe von *Software Defined Radio* (SDR) demonstriert. Dabei ist die Grundidee von SDR, möglichst alle Komponenten eines Übertragungssystems in Software zu realisieren, da dies im Vergleich zum klassischen Hardware-Entwurf höhere Flexibilität und kürzere Entwicklungszeiten verspricht. Nachteilig ist vor allem die geringere Laufzeit-Performance – eine Herausforderung, die bei der Entwicklung des TOoL+-Demonstrator sehr deutlich wurde. Im Rahmen dieser Arbeit wird der Demonstrator daher primär bzgl. der Laufzeit-Performance verbessert. Dazu werden zunächst die beiden zur Vorwärtsfehlerchutzdecodierung verwendeten Algorithmen und anschließend die eingesetzte SDR-Plattform *IfN Generic SDR Toolkit* (IGST) optimiert.

Bei der Optimierung der Algorithmen sind vor allem die in Broadcastsystemen üblichen großen Fehlerschutzcodewortlängen für eine performante Implementierung sehr herausfordern. Zudem stellen diese einen wesentlicher Unterschied dieser Arbeit zu anderen Veröffentlichungen dar. Durch den Einsatz eines geeigneten Speicherlayouts, sowie von Single Instruction Multiple Data (SIMD) und Festkommaarithmetik kann die Performance der beiden Algorithmen um bis zu Faktor 50 gegenüber der Referenzimplementierung gesteigert werden und eine Echtzeitdecodierung ist dadurch möglich. Dabei erweist sich insbesondere der gewählte Intra-Frame-Ansatz von Vorteil, da dadurch der Speicherverbrauch minimiert und der schnelle Cache-Speicher des Hauptprozessors besser genutzt wird.

Verschiedene SDR-Plattformen unterscheiden sich z. T. deutlich bzgl. der Ansätze zur effizienten Datenverarbeitung und es ist unklar, welche Auswirkungen diese auf die Laufzeit-Performance einer SDR-Anwendung haben. Daher werden die SDR-Plattformen zunächst diesbezüglich miteinander verglichen und daraus Anforderungen an eine optimierte Plattform abgeleitet. Da hierfür noch kein Vergleichsbenchmark existiert, wird ein solcher entwickelt und anschließend verwendet. Ein Ergebnis dieses Vergleichs ist, dass der Einsatz von Task-basierten statt der bisher von IGST verwendeten Thread-basierten Nebenläufigkeit für SDR-Anwendungen von Vorteil ist, da dadurch vor allem die Lastverteilung stark vereinfacht wird und moderne Mehrkernprozessoren besser nutzbar sind. IGST wird daher um eine entsprechende Funktion erweitert. Zusätzlich wird auch die von IGST verwendete Block- und Paketstruktur optimiert, um u. a. die generelle Nutzbarkeit der SDR-Plattform zu erhöhen.

Insgesamt wird die Leistungsfähigkeit des TOoL+-Demonstrators um den Faktor 9 erhöht. Zusätzlich sind die verwendeten Ansätze auch von weiteren Algorithmen, Plattformen und Anwendungsfälle nutzbar und stellen somit auch eine Verbesserung von SDR allgemein dar. Zusammenfassend können die Ergebnisse dieser Arbeit somit als ein maßgeblicher Beitrag angesehen werden, um auch zukünftig die Entwicklung und Demonstration von immer komplexeren Übertragungssystemen wie z. B. von 5G per SDR zu ermöglichen.

Abstract

Due to the increasing demand for video services today's cellular mobile networks are close to their capacity limit. Using a broadcast/multicast based transmission and high tower high power sites for this use case is more cost and resource effective than unicast and low tower low power sites. Therefore, the Tower Overlay over LTE-A+ (TOoL+) system was developed which is based on the current mobile communication system LTE-A. Additionally, TOoL+ was demonstrated by using Software Defined Radio (SDR). The basic idea of SDR is to realize a communication system as far as possible in software. Compared to the traditional hardware development this approach is more flexible and allows for shorter development times. The main drawback of SDR is its more limited runtime performance – a challenge that became very clear when developing the TOoL+ demonstrator. Therefore, in this work the demonstrator or more precisely the two algorithms used for the decoding of the forward error correction (FEC) codes and the SDR platform itself (IfN Generic SDR Toolkit – IGST) will be optimized.

Regarding the FEC decoding, especially the long code block sizes typically used by broadcast systems are very challenging for a high-performance implementation. Additionally, this is a main difference of the implementation described in this work to other publications. Amongst other things the optimized implementation uses a suitable memory layout, Single Instruction Multiple Data (SIMD) and fix point arithmetic to realize a speed up by a factor of 50 compared to a reference implementation. The optimized implementation applies an intra frame approach to minimize the required memory footprint and to allow an improved usage of the fast processor's cache. Using this implementation a real time transmission becomes possible.

Various SDR platforms use very different approaches for efficient data processing and it is unclear how these approaches affect the runtime performance. Therefore, these approaches of the SDR platforms are compared and requirements to an optimized platform are derived. Since there is no benchmark for this purpose, one is first developed and then used. One result of this benchmark is, that the usage of task based instead of thread-based concurrency is beneficial for SDR applications as this allows for an improved load distribution on multi core processor units. Therefore, IGST is extended by such functionality. Additionally, the block and packet structure of IGST is improved to increase the usability of the SDR platform significantly.

Altogether, the optimizations presented in this work improve the performance of the TOoL+ demonstrator by a factor of 9. The presented approaches can also be applied on other algorithms, platforms and use cases. Hence, the results of this work can be considered as a significant contribution to improve SDR in general and to develop and demonstrate even more complex communication systems such as 5G by using SDR.

Inhaltsverzeichnis

Abbildungsverzeichnis	xii
Tabellenverzeichnis	xv
Verzeichnis der Codebeispiele	xvi
Verzeichnis der Abkürzungen	xvii
1 Einleitung	1
2 Tower Overlay over LTE-A+ - Ein Beispiel eines terrestrischen Übertragungssystems	5
2.1 Grundlagen der digitalen Signalübertragung	5
2.1.1 Kanalstruktur des Übertragungssystems LTE-A	6
2.1.2 Komponenten der Kanäle MCH und PMCH	7
2.2 Übersicht über die Systemeigenschaften von T0oL+	11
2.2.1 Übersicht über die Erweiterungen von LTE-A zu LTE-A+	12
2.2.2 Systemintegration des LTE-A+-Trägers	13
2.2.3 Kooperative Spektrumsnutzung per DVB-T2-Zeitmultiplex	14
2.2.4 T0oL+-Demonstratoren und -Feldversuche	14
2.3 Performance-kritische Algorithmen der Bitübertragungsschicht von T0oL+	15
2.3.1 Turbo-Decodierung	16
2.3.2 Low-Density-Parity-Check-Decodierung	19
3 Software Defined Radio	22
3.1 Einführung	22
3.2 SDR-Plattformen - Eine Übersicht	24
3.2.1 GNU Radio	24
3.2.2 IfN Generic SDR Toolkit	25
3.2.3 Weitere SDR-Plattformen	25
3.3 SDR-Frontends	28
3.3.1 Architektur der SDR-Frontends	28
3.3.2 Ausgewählte SDR-Frontends	29
3.4 Herausforderungen an SDR-Plattformen und -Anwendungen	30
3.5 Die Architektur des IfN Generic SDR Toolkits	32
3.5.1 Grundkomponenten des IGSTs – Verarbeitungsblöcke	33
3.5.2 Grundkomponenten des IGSTs – Verbindungen	35
3.5.3 Grundkomponenten des IGSTs – Pakete	36
3.5.4 Erzeugung des Modells eines Übertragungssystems	37
3.5.5 Die grafische Benutzeroberfläche	37
4 Evaluation der SDR-Plattform IGST am Beispiel T0oL+	41
4.1 Implementierung von T0oL+ per SDR	41
4.2 Analyse der SDR-Implementierung von T0oL+	44
4.2.1 Performanceanalyse der Implementierung	44
4.2.2 Praktische Analyse der SDR-Plattform IGST	47
4.3 Bewertung von IGST gegenüber den Herausforderungen an SDR-Plattformen und -Anwendungen	51
4.4 Forschungsfragen und Optimierungsbedarf	54

5	Optimierung der SDR-Implementierung Performance-kritischer Algorithmen von T0oL+	56
5.1	Ansätze zur Optimierung von Performance-kritischen Algorithmen	56
5.2	Turbo-Decodierung	59
5.2.1	Implementierung eines Turbo-Decoders per SDR	59
5.2.2	Optimierung der Implementierung	60
5.2.3	Evaluation der vorgeschlagenen Turbo-Decoder-Implementierung	65
5.2.4	Gesamtergebnis der Turbo-Decoder-Optimierung	73
5.3	Low-Density-Parity-Check-Decodierung	74
5.3.1	Implementierung eines LDPC-Decoders per SDR	74
5.3.2	Optimierung der Implementierung	76
5.3.3	Evaluation der vorgeschlagenen LDPC-Decoder-Implementierung	80
5.3.4	Gesamtergebnis der LDPC-Decoder-Optimierung	91
5.4	Zusammenfassung der Ergebnisse	92
6	Vergleich von verschiedenen SDR-Plattformen unter besonderer Berücksichtigung der verwendeten Datenverarbeitungs-Konzepte	94
6.1	Ansätze zur effizienten Datenverarbeitung	94
6.1.1	IFN Generic SDR Toolkit	95
6.1.2	GNU Radio	96
6.1.3	Task-basierte Nebenläufigkeit am Beispiel Intel Threading Building Blocks	97
6.2	Ein Benchmark zum Vergleich des Datendurchsatzes von SDR-Plattformen	100
6.2.1	Szenario 1 - Analyse des Scheduling-Verhaltens und der Skalierbarkeit	100
6.2.2	Szenario 2 - Reale SDR-Anwendung	102
6.3	Analyse des Datendurchsatzes mittels des Benchmarks	103
6.3.1	Evaluation von Szenario 1	103
6.3.2	Evaluation von Szenario 2	105
6.4	Bewertung von GNU Radio und Intel TBB gegenüber den Herausforderungen an SDR-Plattformen und -Anwendungen	107
6.5	Ableitung weiterer Anforderungen an eine optimierte SDR-Plattform	109
6.6	Zusammenfassung der Ergebnisse	110
7	Entwurf einer optimierten SDR-Plattform auf Basis des IGSTs	112
7.1	Ein Konzept zur Verbesserung der Blockstruktur	113
7.1.1	Optimierungen der Nutzbarkeit der Algorithmik	113
7.1.2	Verwendung von Task-basierter Nebenläufigkeit	116
7.2	Ein Konzept zur Verbesserung der Paketstruktur	119
7.2.1	Ansätze zur Paketstrukturierung	119
7.2.2	Entwurf einer optimierten Paketstruktur	120
7.3	Zusammenfassung der Optimierungsmaßnahmen	124
8	Gesamtevaluation der Optimierung	126
8.1	Abschätzung des Portierungsaufwandes	126
8.2	Performanceanalyse der optimierten Implementierung von T0oL+	129
8.3	Bewertung der optimierten SDR-Plattform IGST gegenüber den Herausforderungen an SDR-Plattformen und -Anwendungen	134
8.4	Gesamtfazit	136

9 Zusammenfassung	138
A Anhang	141
A.1 Basis-Klassen des IGSTs	141
A.2 Vollständige Blockdiagramme des TOoL+-Demonstrators	144
A.2.1 Blockdiagramme der SDR-Anwendung LTE-A+ TX	144
A.2.2 Blockdiagramme der SDR-Anwendung LTE-A+ RX	146
A.2.3 Blockdiagramme der SDR-Anwendung DVB-T2 RX	149
Literaturverzeichnis	151
Eigene Publikationen	166
Betreute Abschlussarbeiten	168

Abbildungsverzeichnis

1.1	Prognostiziertes Wachstum des mobilen Datenverkehrs	1
1.2	Architektur des Tower Overlay über LTE-A+	2
2.1	Komponenten eines digitalen Übertragungssystems	5
2.2	Zuordnung der unterschiedlichen Downlink-Kanäle des Übertragungssystems LTE-A	6
2.3	Blockdiagramm der Kanäle MCH und PMCH von LTE-A	7
2.4	Konstellationsdiagramm einer 16-QAM mit Gray-Code	9
2.5	Framing-Struktur von LTE-A bei Verwendung des Framestrukturtyps 1 und des erweiterten Cyclic Prefix	10
2.6	Erweiterung der Cyclic-Prefix-Länge des Übertragungssystems LTE-A zur Unterstützung von HTHP-Sendern und längeren Mehrwegelaufzeiten	12
2.7	Modulation eines hybriden DVB-T2/LTE-A+-Trägers unter Zuhilfenahme von DVB-T2 Future Extension Frames	13
2.8	Übersicht über die Empfangsleistungspegel, welche beim Feldversuch von TOoL+ im Zentrum und süd-östlichen Teil von Paris gemessen wurden	15
2.9	Blockschaltbild der Turbo-Codierung	16
2.10	Trellis-Diagramm des von LTE-A verwendeten Turbocodes	17
2.11	Tanner-Graph eines einfachen LDPC-Codes	19
3.1	Aufteilung der Komponenten eines digitalen Übertragungssystems in Soft- und Hardwareanteile	22
3.2	Anzahl der Veröffentlichungen mit SDR-Bezug seit 1995	23
3.3	IGST-Modellgraph mit Quelle, Verarbeitungsblock und Senke	25
3.4	Klassendiagramm des IGSTs	32
3.5	Sequenzdiagramm eines typischen Modellablaufs	39
3.6	Graphische Oberfläche des IGSTs	40
4.1	Aufbau des TOoL+-Demonstrators	41
4.2	Vereinfachte Blockdiagramme des hybriden Modulators und des LTE-A+- Terminalprototyps des TOoL+-Demonstrators	42
4.3	Spektrum des Zeitmultiplexes aus DVB-T2 und LTE-A+	44
4.4	Performanceanalyse des TOoL+-Demonstrators – (De-)Codierung von 60 Sekunden der TOoL+-Wellenform	45
4.5	Performanceanalyse des TOoL+-Demonstrators – Relative Verarbeitungs- dauer der unterschiedlichen Algorithmen	47
4.6	Typischer Zustandsautomat inklusive der computeThread-Methode eines einfachen Verarbeitungsblocks	48
4.7	Klassendiagramm des in den LTE-A+-Übertragungsmodellen hauptsächlich verwendeten Pakettyps CLTEPhyMchPacket	49
5.1	Übersicht über einige Formen der Parallelität in Softwareanwendungen	56
5.2	SIMD-Registeraufteilung für verschiedene Datentypen und die SIMD-Befehls- saterweiterungen SSE und AVX	57
5.3	Vektor-basierte Berechnung der α -Metrik am Beispiel des von LTE-A ver- wendeten Turbocodes	63
5.4	Einfluss des eingesetzten MAP-Algorithmus auf die Performance der Tur- bodecoderimplementierung	66

5.5	Einfluss der eingesetzten SIMD-Befehlssatzerweiterung auf die Performance der Turbodecoderimplementierung	67
5.6	Schematische Darstellung der CPU-Taktstufen in Abhängigkeit von der verwendeten SIMD-Befehlssatzerweiterung	68
5.7	Einfluss der Speicheroptimierung auf die Performance der Turbodecoderimplementierung am Beispiel des maxLogMAP-Algorithmus und unter Verwendung der SIMD-Befehlssatzerweiterung AVX	68
5.8	Einfluss der Festkommaimplementierung auf die Performance der Turbodecoderimplementierung	69
5.9	Einfluss mehrerer Verarbeitungsthreads auf die Laufzeit-Performance der Turbodecoderimplementierung	70
5.10	Laufzeitvergleich der vorgeschlagenen Turbodecoderimplementierung mit der AFF3CT-Implementierung	72
5.11	Speicherlayout der Kantenvektoren LLR_{BN} und LLR_{CN} sowie die benötigten Zugriffe auf diese bei einer Knotenaktualisierung	75
5.12	Speicherlayout der Kantenvektoren LLR_{BN} und LLR_{CN} beim Verwenden vom SIMD	78
5.13	Struktur der Paritätsmatrix von LDPC-Codes am Beispiel des Übertragungssystems DOCSIS 3.1	79
5.14	Einfluss des eingesetzten LDPC-Decodieralgorithmus und der SIMD-Erweiterung auf die Performance der Referenzimplementierung	80
5.15	Einfluss der Festkommaarithmetik auf die Performance der generischen LDPC-Decoderimplementierung	81
5.16	Einfluss der Festkommaarithmetik auf die Performance der SIMD-LDPC-Decoderimplementierung	82
5.17	Einfluss der Festkommaarithmetik auf die Performance der AVX-LDPC-Decoderimplementierung auf Plattform 3	83
5.18	Einfluss der Code-abhängigen Optimierungen auf die Performance der LDPC-Decoderimplementierung	85
5.19	Einfluss mehrerer Verarbeitungsthreads auf die Performance der LDPC-Decoderimplementierung	87
5.20	Laufzeitvergleich der vorgeschlagenen LDPC-Decoderimplementierung mit der GNU Radio-Implementierung	88
6.1	Teil des Zustandsautomaten des von GNU Radio verwendeten Schedulers	96
6.2	Blockschaltbild des in Szenario 1 verwendeten Modellgraphens	100
6.3	Klassendiagramm des im SDR-Benchmark verwendeten Klassenkonzeptes	101
6.4	Blockschaltbild der in Szenario 2 verwendeten SDR-Anwendung	102
6.5	Leistungsfähigkeit der verschiedenen SDR-Plattformen für Szenario 1	104
6.6	CPU-Auslastung der SDR-Plattform IGST, Atomic für Szenario 1	105
6.7	Benötigte Verarbeitungszeit und durchschnittliche CPU-Auslastung der verschiedenen SDR-Plattformen bzw. Ansätze für das Szenario 2	106
7.1	Zustandsautomat inklusiver der computeThread- und der process-Methode eines optimierten einfachen asynchronen Verarbeitungsblocks	113
7.2	Klassendiagramm der optimierten Blockoberklassen	115
7.3	Klassendiagramm für das Verbinden der Plattformen IGST und Intel TBB	117
7.4	Klassendiagramm der verbesserten Paketstruktur für SDR-Anwendungen	123

8.1	Blockschaltbild eines generischen Verarbeitungsblocks mit mehreren syn- chronen Eingängen	127
8.2	Blockschaltbild eines generischen Verarbeitungsblocks mit mehreren asyn- chronen Eingängen und einem internen Zustandsautomaten am Beispiel des <i>Resamplings</i>	128
8.3	Interner Zustandsautomat des portierten Blocks Resampling	129
8.4	Gesamtevaluationsergebnis der verschiedenen SDR-Plattformen für das Modell <i>LTE-A+ RX</i>	130
8.5	Performancanalyse des optimierten T _{OO} L+-Demonstrators – Relative Ver- arbeitungsdauer der unterschiedlichen Algorithmen	132
8.6	Evaluation der benötigten Verarbeitungszeit der verschiedenen SDR-Platt- formen für das Modell <i>LTE-A+ RX</i> und unter Berücksichtigung einer zusätzlichen Optimierung des Resamplings	133
A.1	Klassendiagramm der Klasse CGenericProcessingBlock des IGSTs	141
A.2	Klassendiagramm der Klasse CBaseProcessingBlock des IGSTs	141
A.3	Klassendiagramm der Klasse CProcessingBlock des IGSTs	142
A.4	Klassendiagramm der Klasse CSourceBlock des IGSTs	142
A.5	Klassendiagramm der Klasse CSinkBlock des IGSTs	142
A.6	Klassendiagramm der Klasse CUnidirectionalLink des IGSTs	142
A.7	Klassendiagramm der Klasse CPacket des IGSTs	143
A.8	Blockdiagramm der SDR-Anwendung T _{OO} L+ TX – Teil 1	144
A.9	Blockdiagramm der SDR-Anwendung T _{OO} L+ TX – Teil 2	145
A.10	Blockdiagramm der SDR-Anwendung LTE-A+ RX – Teil 1	146
A.11	Blockdiagramm der SDR-Anwendung LTE-A+ RX – Teil 2	147
A.12	Blockdiagramm der SDR-Anwendung LTE-A+ RX – Teil 3	148
A.13	Blockdiagramm der SDR-Anwendung DVB-T2 RX – Teil 1	149
A.14	Blockdiagramm der SDR-Anwendung DVB-T2 RX – Teil 2	150

Tabellenverzeichnis

2.1	Übersicht der modifizierten OFDM-Parameter von LTE-A+	13
3.1	Übersicht über die Hardwarespezifikationen ausgewählter SDR-Frontends .	30
3.2	Übersicht über die möglichen HF-Parameter ausgewählter SDR-Frontends und USRP-Daughterboards	30
3.3	Übersicht der verschiedenen Verarbeitungsbloctypen, ihrer Portanzahl sowie Oberklassen	35
4.1	Übertragungsparameter des hybriden TOoL+-Signals	43
4.2	Analysierte Übertragungsmodelle	45
4.3	Zusammenfassung der Anforderungen an eine optimierte SDR-Plattform .	50
4.4	Bewertung der SDR-Plattform IGST bzgl. der Herausforderungen an SDR- Plattformen und -Anwendungen	53
5.1	Anzahl der Lade- bzw. Speicheroperationen eines naiven bzw. optimierten Turbo-Decoders	61
5.2	Übersicht der für die Evaluation verwendeten Rechner-Plattformen	65
5.3	Übersicht über die CPU-Taktstufen pro Kern in Abhängigkeit von der SIMD-Befehlssatzerweiterung und der CPU-Auslastung	68
5.4	Durchsatzvergleich verschiedener SDR-basierter LDPC-Decoder-Implemen- tierungen	90
6.1	Übersicht der verwendeten SDR-Plattformen und deren Ansätze zur effzi- enten Datenverarbeitung	99
6.2	Bewertung der Plattformen IGST, GNU Radio und Intel TBB bzgl. der Herausforderungen an SDR-Plattformen und -Anwendungen	109
6.3	Weitere Anforderungen an eine optimierte SDR-Plattform aufgrund der Ergebnisse des SDR-Plattformvergleichs	110
7.1	Gesammelte Anforderungen an eine optimierte SDR-Plattform	112
7.2	Ansätze zur Paketstrukturierung innerhalb von SDR-Anwendungen sowie deren Vor- und Nachteile	121
7.3	Bewertung der verschiedenen Ansätze zur Paketstrukturierung innerhalb von SDR-Anwendungen bzgl. der definierten Anforderungen	122
8.1	Bewertungen der optimierten SDR-Plattform IGST bzgl. Herausforderungen an SDR-Plattformen und -Anwendungen	135

Verzeichnis der Codebeispiele

3.1	Codebeispiel eines einfachen Verarbeitungsblocks zur Normierung von Basisbanddaten	34
3.2	Codebeispiel eines Paketformats zum Austausch von IQ-Basisbanddaten mit Hilfsfunktion zur Normierung	36
3.3	Codebeispiel des Modells eines Übertragungssystems	38
5.1	Pseudocode der Funktion <code>computeTurboDecode</code>	60
5.2	Pseudocode der Funktion <code>computeMAP</code>	61
5.3	Pseudocode der Funktion <code>computeMAP</code> mit Schleifenfusion	62
5.4	Pseudocode der Funktion <code>computeLDPCDecode</code>	74
5.5	Pseudocode einer naiven Implementierung von <code>bnUpdate</code>	76
5.6	Pseudocode der optimierten Implementierung von <code>bnUpdate</code>	77
7.1	Vereinfachte Implementierung der Methode <code>TBBWSerialAsyncProcessingBlock::operator()</code>	118
7.2	Codebeispiel eines Blocks, der die <code>DCPacket</code> -Schnittstelle nutzt	123

Verzeichnis der Abkürzungen

3GPP	3rd Generation Partnership Project
A/D	Analog/Digital
ADC	Analog Digital Conversion
AFF3CT	A Fast Forward Error Correction Tool
ALU	Arithmetic Logic Unit
API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
ATSC 3.0	Advanced Television Systems Committee Version 3.0
AVX	Advanced Vector Extensions
AWGN	Additive White Gaussian Noise
BCCH	Broadcast Control Channel
BCH	Broadcast Channel
BCJR	Bahl, Cocke, Jelinek und Raviv
BER	Bit Error Rate
BN	Bit Node
CA	Carrier Aggregation
CCCH	Common Control Channel
CFI	Control Format Indicator
CFS	Completely Fair Scheduler
CGRAN	Comprehensive GNU Radio Archive Network
CN	Check Node
CORBA	Common Object Request Broker Architecture
CP	Cyclic Prefix
CPH	Control Processor Host
CPU	Central Processing Unit
CR	Cognitive Radio
CRC	Cyclic Redundancy Check
D/A	Digital/Analog
DAC	Digital Analog Conversion
DAB+ LSI	Digital Audio Broadcasting+ Local Service Insertion
DCCH	Dedicated Control Channel
DCI	Downlink Control Information

DL-SCH	Downlink Shared Channel
DOCSIS 3.1	Data Over Cable Service Interface Specification Version 3.1
DTAPI	DekTec Application Programming Interface
DTCH	Dedicated Traffic Channel
DVB-C2	Digital Video Broadcasting Cable - 2nd Generation
DVB-S2	Digital Video Broadcasting Satellite - 2nd Generation
DVB-T2	Digital Video Broadcasting Terrestrial - 2nd Generation
eMBB	enhanced mobile broadband
eMBMS	evolved Multimedia Broadcast Multicast Services
eNodeB	Evolved Node B
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FEF	Future Extension Frame
feMBMS	further evolved Multimedia Broadcast Multicast Service
FFT	Fast Fourier Transformation
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FPU	Floating Point Unit
GFLOPS	Giga Floating Point Operations Per Second
GI	Guard Interval
GPP	General Purpose Processor
GPU	Graphic Processing Unit
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HARQ	Hybrid Automatic Repeat Request
HEVC	High Efficiency Video Coding
HF	Hochfrequenz
HI	HARQ Indicator
HT	Hyper Threading
HTHP	High Tower High Power
HPC	High Performance Computing
IBC	International Broadcasting Convention

IF	Intermediate Frequency
IfN	Institut für Nachrichtentechnik
IGST	IfN Generic SDR Toolkit
IoT	Internet of Things
IPTV	Internet Protocol Television
ISI	Inter Symbol Interferenz
ISI	Intersymbolinterferenz
JPEG	Joint Photographic Experts Group
JTNC	Joint Tactical Networking Center
KUAR	Kansas University Agile Radio
LAN	Local Area Network
LDPC	Low Density Parity Check
LLR	Log Likelihood Ratio
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
LTE-A+	LTE Advanced+
LTLP	Low Tower Low Power
MAC	Media Access Control
MAP	Maximum a Posteriori
MCCH	Multicast Control Channel
MCH	Multicast Channel
MIMO	Multiple Input Multiple Output
MIT	Massachusetts Institute of Technology
MME	Mobility Management Entity
MMX	Multi Media Extension
MNO	Mobile Network Operator
MTCH	Multicast Traffic Channel
NAB	National Association of Broadcasters
NB-IoT	NarrowBand IoT
NGMN IC&E	Next Generation Mobile Networks Industry Conference & Exhibition
OAI	OpenAirInterface
OFDM	Orthogonal Frequency Division Multiplexing

OOT	Out-of-Tree
OSA	OAI Software Alliance
OSSIE	Open Source SCA Implementation::Embedded
PBCH	Physical Broadcast Channel
PC	Personal Computer
PCCH	Paging Control Channel
PCFICH	Physical Control Format Indicator Channel
PCH	Paging Channel
PCIe	Peripheral Component Interconnect Express
PDCCH	Physical Downlink Control Channel
PDSCH	Physical Downlink Shared Channel
PHICH	Physical Hybrid ARQ Indicator Channel
PMCH	Physical Multicast Channel
POD	Plain Old Data Structure
PSCH	Primary Synchronization Channel
PSPECTRA	Parallel SPECTRA
PSTL	Parallel Standard Template Library
QAM	Quadrature Amplitude Modulation
QC-LDPC	Quasi Cyclic Low Density Parity Check
QPSK	Quadrature Phase Shift Keying
RAM	Random Access Memory
RCB	Radio Control Board
RFNoC	RF Network on Chip
RLC	Radio Link Control
RoD	Redundancy on Demand
SCA	Software Communication Architecture
SC-PTM	Single Cell Point To Multipoint
SDR	Software Defined Radio
SFN	Single Frequency Network
SIM	Subscriber Identity Module
SIMD	Single Instruction Multiple Data
SiMoNe	Simulator for Mobile Networks

SISO	Soft-in Soft-out
SMT	Simultaneous Multithreading
SNR	Signal to Noise Ratio
SOVA	Soft-Output-Viterbi-Algorithmus
SPECTRA	Signal Processing Environment for Continues Real-time Applications
SSCH	Secondary Synchronization Channel
SSE	Streaming SIMD Extensions
Intel TBB	Intel Threading Building Blocks
TBBW	Intel-TBB-Wrapper
TCC	Turbo-Convolutional-Code
TCP/IP	Transmission Control Protocol/Internet Protocol
TDD	Time Division Duplex
TDP	Thermal Design Power
TOoL+	Tower Overlay over LTE-A+
TPC	Turbo-Product-Code
UE	User Equipment
UHD	Ultra High Definition
UHF	Ultra High Frequency
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VoD	Video on Demand
VOLK	Vector-Optimized Library of Kernels
WLAN	Wireless Local Area Network
XML	Extensible Markup Language
ZF	Zwischenfrequenz