Marcos B.S. Tavares

On Low-Density Parity-Check Convolutional Codes: Constructions, Analysis and VLSI Implementation

Beiträge aus der Informationstechnik

Mobile Nachrichtenübertragung

Nr. 48

Marcos Bruno Saldanha Tavares

On Low-Density Parity-Check Convolutional Codes:

Constructions, Analysis and VLSI Implementation

VOGT

Dresden 2010

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Bibliographic Information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibiograpic data is available in the internet at http://dnb.ddb.de.

Zugl.: Dresden, Techn. Univ., Diss., 2010

Die vorliegende Arbeit stimmt mit dem Original der Dissertation „On Low-Density Parity-Check Convolutional Codes: Constructions, Analysis and VLSI Implementation“ von Marcos B.S. Tavares überein.

Gesetzt vom Autor

ISBN 978-3-938860-38-0

Jörg Vogt Verlag
Niederwaldstr. 36
01277 Dresden
Germany

Phone: +49-(0)351-31403921
Telefax: +49-(0)351-31403918
e-mail: info@vogtverlag.de
Internet : www.vogtverlag.de

TECHNISCHE UNIVERSITÄT DRESDEN

On Low-Density Parity-Check Convolutional Codes:

Constructions, Analysis and VLSI Implementation

Marcos Bruno Saldanha Tavares

von der
Fakultät Elektrotechnik und Informationstechnik
der Technischen Universität Dresden

zur Erlangung der akademischen Grades eines

Doktoringenieurs
(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender:	Prof. Dr.-Ing. habil. Adolf Finger
Gutachter:	Prof. Dr.-Ing. Gerhard P. Fettweis
	Prof. Dr. Kamil Sh. Zigangirov

Tag der Einreichung:	11.01.2010
Tag der Verteidigung:	22.02.2010

To the most important persons in my life:

Adriana,

Joaquim Gothardo,

Dona Zélia,

Seu Marcos,

Lúcio Henrique,

Júlio César,

Juliana.

And to all my teachers and companions in thought.

Abstract

This dissertation concerns the study and further development of *low-density parity-check (LDPC) convolutional codes*. In contrast to LDPC block codes, LDPC convolutional codes are not limited to a single block length, and the same encoder and decoder apparatuses can be used for processing varying block lengths.

We follow the philosophy of *probabilistic coding*. Therefore, our main objective is to find codes that optimize the performance/complexity tradeoff. In this sense, because the same encoders/decoders can be used for processing different block lengths (which may imply a constant complexity for different performances), the LDPC convolutional codes have the potential of achieving good performance/complexity tradeoffs.

One of the main inventions presented in this dissertation are the tail-biting versions of LDPC convolutional codes. The *tail-biting LDPC convolutional codes* represent an elegant solution to the rate-loss problem of terminated code sequences while still keeping the same underlying bipartite graph structures and, thus, the same performance/complexity properties as their mother convolutional codes. We show with the help of several computer simulations that the tail-biting LDPC convolutional codes are serious competitors to conventional LDPC block codes.

Two different construction approaches for LDPC convolutional codes and their tail-biting versions are proposed. The first construction approach is of algebraic nature and produces time-invariant codes. The second construction approach uses performance-optimized graph templates called *protographs* as base for the generation of larger graphs. Along with these two construction procedures, we also provide bounds to the performance of codes with quasi-cyclic symmetries. Moreover, we study the bipartite graph configurations that lead to the error-floor phenomenon and propose techniques for avoiding these harmful graph configurations during the code construction. By means of computer simulations, we compare the performance of the codes obtained from the procedures discussed above against codes from constructions presented in the literature, where we can observe the superiority of our constructions in terms of the performance/complexity tradeoff.

In order to assess the performance of code ensembles – rather than the performance of a single code – we propose a framework for the asymptotic analysis of free/mininum distances and trapping set properties of protograph-based ensembles of LDPC convolutional codes and their tail-biting versions. Using numerical analysis, we show the dual nature of tail-biting LDPC convolutional codes, which can behave as block or as convolutional codes depending on the length of the tail-biting code sequences being considered. Moreover, in the situations where the tail-biting LDPC convolutional codes behave as block codes, we show that the performance can be improved while the encoding/decoding complexity remains constant.

Especially for low coding rates – where the complexity of belief propagation decoding becomes high – we present a new protograph-based code construction method based on the braided concatenation of component convolutional codes. The codes originating from this construction, which we call *braided protograph convolutional codes*, can be decoded using the belief propagation algorithm or a trellis based decoding algorithm, depending

on how the underlying protographs are lifted. Hence, the decoding algorithm for the braided protograph convolutional codes can be chosen such that the decoding complexity is minimized. Computer simulations show the superiority (in the moderate to high SNR region) of the tail-biting braided protograph convolutional codes when compared against turbo codes with the same length, and with component encoders with the same number of states. Moreover, the interleaving structure of the braided protograph convolutional codes and their tail-biting versions enables the development of efficient, high-speed decoding architectures. Furthermore, an asymptotic analysis of the convergence properties of the braided protograph convolutional codes confirms their clear superiority to the turbo codes.

Coding for the multiple-access channel is also addressed in this dissertation. For this purpose, we introduce a new scheme called *braided code division multiple access (BCDMA)*, which is based on *orthogonal frequency division multiplexing (OFDM)* and braided convolutional codes. The BCDMA scheme represents a paradigm shift in the design of modems for multiple-access applications. In this case, the tasks of error-correction coding, interleaving for channel diversity exploitation, modulation and multiple access – which are generally processed by separate elements of a modem – are now concentrated in one single entity. Moreover, the comparison of the BCDMA scheme against conventional *bit-interleaved coded modulation (BICM)* schemes based on turbo and LDPC block codes shows that the BCDMA scheme is an attractive option in terms of the performance/complexity tradeoff.

In order to be able to evaluate the performance/complexty tradeoff in its fundamentals, we study VLSI decoding architectures for LDPC convolutional codes and their tail-biting versions. For this purpose, we propose an algebraic framework that captures the structured parallelism inherent to the bipartite graphs underlying the LDPC convolutional codes and their tail-biting versions. From this algebraic framework, we derive an architectural template that can be used as base to form even more powerful decoders. In this case, a new dimension of scalability is obtained, which is not possible for LDPC block codes. As case studies, we consider the implementation of programmable, high-speed decoding processors. Synthesis results and a chip implementation show that it is possible to achieve throughputs in the range of Gbit/s/Iteration, still with relatively small area and low power consumption.

Kurzfassung

Diese Dissertation beschreibt die Untersuchung und die Weiterentwicklung von *Low-Density Parity-Check (LDPC) Faltungscodes*. Im Unterschied zur LDPC-Blockcodes sind LDPC-Faltungscodes nicht auf eine bestimmte Blocklänge beschränkt, und die gleichen Encoder- und Decoder-Vorrichtungen können benutzt werden, um variable Blocklängen zu verarbeiten.

Wir verfolgen das Konzept der *probabilistischen Kodierung*. Unsere Hauptaufgabe ist die Suche nach Codes, die den Kompromiss zwischen Leistungsfähigkeit und Komplexität optimieren. Da die gleichen Encoder und Decoder für die Verarbeitung von verschiedenen Blocklängen genutzt werden können (und dies konstante Komplexität für variable Leistungsfähigkeit bedeutet), besitzen LDPC-Faltungscodes das Potenzial, gute Kompromisse zwischen Leistungsfähigkeit und Komplexität zu erzielen.

Eine der Haupterfindungen, die in dieser Dissertation vorgestellt wird, sind die Tail-Biting Versionen der LDPC-Faltungscodes. Die *Tail-Biting LDPC-Faltungscodes* stellen eine elegante Lösung des Rate-Loss Problems von terminierten Codesequenzen dar. Sie besitzen die grundlegende zweiteilige Graphenstrukturen zusätzlich zum Kompromiss zwischen Leistungsfähigkeit und Komplexität ihrer Mutter-Faltungscodes. Wir zeigen mit der Hilfe von verschiedenen Computersimulationen, dass die Tail-Biting LDPC-Faltungscodes ernstzunehmende Konkurrenten von herkömmlichen LDPC-Blockcodes sind.

Zwei verschiedene Konstruktionsansätze für LDPC-Faltungscodes und ihre Tail-Biting Versionen werden vorgeschlagen. Der erste Konstruktionsansatz ist von algebraischen Natur und erzeugt zeitinvariante Codes. Der zweite Konstruktionsansatz nutzt optimierte Graphenvorlagen (bekannt als *Protographen*) als Basis für die Eurzeugung von großeren Graphen. Neben diesen zwei Konstruktionsprozeduren werden auch Grenzen der Leistungsfähigkeit von Codes mit quasi-zyklischen Symmetrien aufgezeigt. Darüber hinaus untersuchen wir die Graphenkonfigurationen, die zum Error-Floor Phänomen führen, und wir schlagen Techniken zur Vermeidung dieser schlädlichen Graphenkonfigurationen während der Codekonstruktion vor. Anhand von Computersimulationen vergleichen wir die Leistungsfähigkeit der Codes, die mit den oben erwähnten Prozeduren erzeugt wurden, mit Codes von Konstruktionen, die in der Literatur präsentiert werden, wobei wir die Überlegenheit unserer Codes bezüglich des Kompromisses zwischen Leistungsfähigkeit und Komplexität nachweisen können.

Mit der Absicht die Leistungsfähigkeit von Codeensembles – anstatt der Leistungsfähigkeit von einem einzelnen Code – zu bewerten, schlagen wir vor, ein Rahmenwerk für die asymptotische Analyse der Frei- und Minimaldistanz und auch der Trapping-Set Eigenschaften von protograph-basierten Ensembles von LDPC-Faltungscodes und ihrer Tail-Biting Versionen zu untersuchen. Anhand von numerischer Analyse wird die duale Natur von Tail-Biting LDPC-Faltungscodes festgestellt, die sich abhängig von ihren Längen entweder als Blockcodes oder als Faltungscodes verhalten. Ferner wird es gezeigt, dass sich die Leistungsfähigkeit in Situationen, in denen Tail-Biting LDPC-Faltungscodes als Blockcodes arbeiten, ohne Änderung der Encodierungs-/Decodierungskomplexität verbessern lässt.

Besonders für niedrige Codierungsraten – wenn die Komplexität der Belief-Propagation Decodierung steigt – präsentieren wir eine neuartige protograph-basierte Codekonstruktionsmethode, die sich auf der umflochtenen Verkettung von Komponentenfaltungscodes basiert. Die Codes, die bei dieser Konstruktionsmethode entstehen, bezeichnen wir als *Braided Protograph Faltungscodes*. Die Braided Protograph Faltungscodes können abhängig von der Weise, auf der die grundlegenden Protographen zusammengefügt werden, mittels der Belief-Propagation oder Trellis-basierten Algorithmen decodiert werden. Infolgedessen kann der Decodierungsalgorithmus für Braided Protograph Faltungscodes so gewählt werden, dass die Decodierungskomplexität minimiert wird. Computersimulationen zeigen die Überlegenheit (in der moderaten bis hohen SNR-Region) der Tail-Biting Braided Protograph Faltungscodes, wenn diese mit Turbo-Codes der gleichen Blocklänge und der gleichen Anzahl von Zuständen des Komponentencodes verglichen werden. Darüber hinaus ermöglicht die Interleaving-Struktur von Braided Protograph Faltungscodes und von ihren Tail-Biting Versionen die Entwicklung von effizienten Hochgeschwindigkeitsdecoder-Architekturen. Zudem bestätigt die asymptotische Analyse der Konvergenzeigenschaften der Braided Protograph Faltungscodes ihre eindeutige Überlegenheit gegenüber Turbo-Codes.

Codierung für den Vielfachzugriffskanal wird zusätzlich in dieser Dissertation betrachtet. Für diesen Zweck wird ein neuartiges Schema entwickelt, das wir *Braided Code Division Multiple Access (BCDMA)* nennen und *Orthogonal Frequency Division Multiplexing (OFDM)* und Braided Faltungscodes als Grundlagen besitzt. Das BCDMA-Schema stellt einen Paradigmenwechsel für den Entwurf von Modems für Vielfachzugriffsanwendungen dar. In diesem Fall werden die Aufgaben der Kodierung für Fehlerkorrektur, Interleaving für die Ausnutzung der Kanaldiversität, Modulation und Vielfachzugriff, die normalerweise von verschiedenen Elementen eines Modems übernommen werden, in einer einzigen Einheit vereinigt. Darüber hinaus zeigt ein Vergleich zwischen dem BCDMA-Schema und herkömmlichen Turbo-Code-basierten *Bit-Interleaved Coded Modulation (BICM)* Schemata die Atraktivität des BCDMA-Schemas im Hinblick auf den Kompromiss zwischen Leistungsfähigkeit und Komplexität.

Mit dem Ziel den Kompromiss zwischen Leistungsfähigkeit und Komplexität in seinen Grundlagen zu evaluieren haben wir die VLSI-Decoderarchitekturen für LDPC-Faltungscodes und ihre Tail-Biting Versionen untersucht. Für diesen Zweck schlagen wir ein algebraisches Rahmenwerk vor, das den Parallelismus inhärent zu den grundlegenden zweiteiligen Graphen der LDPC-Faltungscodes und ihrer Tail-Biting Versionen erfasst. Anhand dieses algebraischen Rahmenwerks wird eine architektonische Vorlage hergeleitet, die benutzt werden kann, um noch leistungsfähigere Decoder zu bauen. In diesem Fall ist eine neue Dimension der Skalierbarkeit erreicht, die mit LDPC-Blockcodes nicht möglich ist. Als Fallbeispiele betrachten wir die Implementierung von programmierbaren Hochgeschwindigkeitsdecoderprozessoren. Ergebnisse der Synthese und der Chip-Implementierung zeigen, dass Datendurchsätze in der Größenordnung von Gbit/s/Iteration mit relativ kleiner Fläche und niedrigem Leistungsverbrauch erreichbar sind.

Acknowledgements

I would like to express my sincere gratitude to Prof. Gerhard P. Fettweis for giving me the opportunity and the necessary resources to pursue this research work in the Vodafone Chair. I am also thankful and proud of the very fruitful collaborations and discussions with remarkable individuals from the Vodafone Chair and from other institutions. Especially, I would like to acknowledge:

- Prof. Kamil Sh. Zigangirov
- Dr. Emil Matúš
- Dr. Michael Lentmaier
- Dipl.-Ing. Marcel Bimberg
- Dipl.-Ing. Steffen Kunze

Furthermore, I am deeply indebted to my dear parents, my amazing siblings, my charming wife Adriana and my wonderful little boy Joaquim, who unconditionally support me and give me strength in all my enterprises.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ACE	Approximate Cycle Extrinsic message degree
AGU	Address Generation Unit
ALU	Arithmetic Logic Unit
APP	A Posteriori Probability
ASIC	Application-Specific Integrated-Circuit
ASIP	Application-Specific Instruction-Set Processor
AWGN	Additive White Gaussian Noise
BICM	Bit-Interleaved Coded Modulation
BCDMA	Braided Code Division Multiple Access
BCJR	Bahl-Cocke-Jelinek-Raviv
BEC	Binary Erasure Channel
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CMOS	Complementary Metal-Oxide-Semiconductor
DMEM	Data Memory
DVB-S2	Digital Video Broadcasting - Satellite - Second Generation
DSP	Digital Signal Processor
EMD	Extrinsic Message Degree
EXIT	EXtrinsic Information Transfer
FER	Frame Error Rate
FPGA	Field Programmable Gate Array
FU	Functional Unit
GOPS	Giga Operations Per Second
IMEM	Instruction Memory
LDPC	Low-Density Parity-Check

LLR	Log-Likelihood Ratio
MCP	Multiple Convolutional Permutor
MMSE	Minimum Mean-Square Error
MPSoC	Multi-Processor System-on-Chip
OFDM	Orthogonal Frequency Division Multiplexing
PDF	Probability Density Function
PCU	Programm Control Unit
PEG	Progressive Edge-Growth
QPSK	Quadrature Phase Shift Keying
QAM	Quadrature Amplitude Modulation
RAM	Random-Access Memory
RISC	Reduced Instruction Set Computing
ROM	Read-Only Memory
SDR	Software Defined Radio
SISO	Soft-Input/Soft-Output
SIMD	Single Instruction, Multiple Data
STA	Synchronous Transfer Architecture
SNR	Signal-to-Noise-Ratio
VLIW	Very Long Instruction Words
VLSI	Very Large Scale of Integration
XOR	eXclusive-OR
WiMAX	Worldwide Interoperability for Microwave Access

List of Frequently Used Symbols and Operators

Symbols

b	Number of input in a convolutional encoder
c	Number of outputs in a convolutional encoder
δ_D	Decoding delay
D	Delay operator
d_{free}	Free distance
$d_{\min}$	Minimum distance
$d_{\text{free}}^{\text{ts}}(\Delta)$	Minimum free Δ-trapping set size
$d_{\min}^{\text{ts}}(\Delta)$	Minimum Δ-trapping set size
$\mathbb{F}$	Finite field
$\mathbb{F}_q$	Finite field with characteristic q
e	2.71828183...
$G = (V, C, E)$	Bipartite (Tanner) graph: V is the set of variable nodes, C is the set of check nodes and E is the set of edges
$\mathbf{H}_{[0,\infty]}^{\text{T}}$	Syndrome former matrix of an LDPC convolutional code
$\tilde{\mathbf{H}}_{[0,t_N]}$	Parity-check matrix of a tail-biting LDPC convolutional code for t_N time instants
I	Number of decoding iterations or number of processors in a pipeline decoder
J	Variable node degree of a regular LDPC code
K	Check node degree of a regular LDPC code
m_g	Encoder memory of a convolutional code
m_s	Syndrome former memory of an LDPC convolutional code
ν_s	Constraint length of an LDPC convolutional code
N	Block length of a code convolutional code

π	3.14159265...
t	Time instant
t_N	Number of time instants
T	Period
R	Coding rate
$\mathbb{R}$	Set of real numbers
$s_{\min}$	Minimum stopping set size
$\mathbb{Z}$	Set of integers
$\mathbb{Z}^+$	Set of positive integers: $1, 2, 3, \cdots$
$\mathbb{Z}^*$	Set on nonnegative integers: $0, 1, 2, 3, \cdots$

Operators

$(\cdot)^{\mathrm{T}}$	Transpose of a matrix or vector
$(\cdot)^{-\mathrm{T}}$	Inverse and Transpose of a matrix
$\lvert \cdot \rvert$	Cardinality of a set
$[\mathbf{M}]_{a:b,c:d}$	Submatrix of $\mathbf{M}$ with elements from the rows a to b and from the columns c to d
$[\mathbf{v}]_{a:b}$	Subvector of $\mathbf{v}$ with elements from positions a to b
$\mathrm{abs}(\cdot)$	Absolute value of a variable
$\deg[\cdot]$	Degree of a check node or variable node
$\dim(\cdot)$	Dimension of a vector
$\mathrm{EMD}(\cdot)$	Extrinsic message degree of a set of variable nodes
$\mathcal{H}(\cdot)$	Entropy function
$\mathcal{W}_H[\cdot]$	Returns the Hamming weight of a sequence
$\mathcal{W}_H^{\mathrm{P}}[\cdot]$	Returns the number of nonzero coefficients of a polynomial
$\mathcal{W}_H^{\mathrm{M}}[\cdot]$	Returns a weight matrix, whose entries are the number of nonzero coefficients of the polynomials that are entries of the original polynomial matrix

1

Introduction

We currently live in the *information age*: we are able to communicate with each other, to transfer information and to have access to knowledge almost anytime and anywhere. From a socio-economic viewpoint, the arrival of the information age was triggered by an *industrial revolution*, which was based on a new kind of industry that is concerned with the manipulation of information.

Analogous to the industrial revolution of the 18-th century, the industrial revolution associated with the information age has its origins in technological breakthroughs. One of the masterpieces that gave important impulses to the information age is the work that Claude E. Shannon presented in [Sha48]. The main contribution of this seminal work was to recognize the *bit* as the fundamental entity behind a communication process. This simplistic recognition has led to new paradigms in the design and analysis of communication systems that allowed unprecedented technological advances, which are present in our society in the form of Internet, wireless communications, etc.

In this context, the definition of communication given by Shannon is as follows:

Definition 1.1:

> *"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point."*
>
> C.E. Shannon.

□

From the definition above, we can deduce the existence of a medium, where the information (messages) between two points is conveyed. This medium (or communication channel) generally inserts interference in the message being transmitted, such that a mechanism to recover the original message becomes necessary. This message recovering mechanism is the main topic addressed by this dissertation and is called *channel coding*.

The history of channel coding also starts in [Sha48] with the *channel coding theorem*, which states that reliable communication is achievable if the transmission rate is smaller than a parameter called *channel capacity*. In order to achieve reliable communication, redundancy is added in a controlled way to the original message $\mathbf{u}$ in a process called *channel encoding*. The outcome of the channel encoding process is the encoded message $\mathbf{v}$ that is transmitted through the communication channel and, due to interference, is modified to the received message $\mathbf{r}$. The receiving party then tries to recover the original message $\mathbf{u}$ from the received message $\mathbf{r}$ in a process called *channel decoding*.

In this sense, the algorithms behind the channel encoding and decoding processes are derived from mathematical constructions called *channel codes*. The first channel codes – Hamming codes – were also introduced in [Sha48]. Because there was still room for plenty of improvements, mathematicians and engineers were faced with the problem of devising practical ways to communicate reliably with data rates next to the channel capacity. For this purpose, several channel coding schemes have been proposed over the last decades. The reader interested in the historic development of channel coding is referred to the survey paper by Costello and Forney [CF07].

Amongst the several approaches to devise channel coding schemes, *probabilistic coding* has led to the most important discoveries. The idea behind probabilistic coding is to find classes of channel codes that optimize the tradeoff between performance and encoding/decoding complexity. The first members of the class of probabilistic codes are the convolutional codes proposed by Elias [Eli55]. Moreover, the further development of channel coding schemes composed by convolutional codes[1] culminated with the invention of the *turbo codes* by Berrou, Glavieux and Thitimajshima [BGT93]. The turbo codes were the first channel coding scheme that enabled reliable communication with data rates very close to the channel capacity still with modest encoding and decoding complexity. The main idea behind the low-complexity decoding of turbo codes is to iteratively perform the decoding of the received message $\mathbf{r}$ using low-complexity convolutional decoders until the convergence to the original message $\mathbf{u}$ is achieved. In the same way, the encoding process has also a low complexity, since it is also performed by simple convolutional encoders.

The invention of the turbo codes started a revolution that caused the rediscovery of Gallager's *low-density parity-check (LDPC) codes* [Gal63] by MacKay [MN95,MN96] and Spielman [SS96, Spi96]. Due to their near-channel-capacity performance and low-complexity iterative decoding algorithms, the LDPC codes are currently seen as serious competitors to turbo codes in practical applications. Actually, Wiberg showed in his doctoral dissertation [Wib96] that turbo codes and LDPC codes could be interpreted as instances of codes defined on sparse graphs, and that their decoding algorithms could be understood as instances of a decoding algorithm called *sum-product algorithm*. The results presented by Wiberg together with previous results presented by Tanner in [Tan81] (and also rediscovered by Wiberg in [Wib96]) founded a new field within the channel coding research called *codes defined on graphs*, which is the current state of the art.

1.1 Objectives and Outline of this Dissertation

This dissertation deals with codes defined on graphs. More specifically, we will focus on the convolutional counterparts of Gallager's LDPC block codes called *LDPC convolutional codes*, which were introduced by Jiménez Feltström and Zigangirov in [FZ99]. In contrast to the LDPC block codes, the LDPC convolutional codes are not limited to a single message length, and the same encoder and decoder circuits can be used for processing diverse message lengths.

In the light of the probabilistic coding philosophy, we are interested in finding new constructions of LDPC convolutional codes that optimize the tradeoff between performance and complexity. In this sense, the fact that the same encoder/decoder can be used for encoding/decoding several message lengths makes the LDPC convolutional codes potential candidates for achieving good performance/complexity tradeoffs. In order to be able to assess the performance/complexity tradeoff of our constructions, we also present several theoretical analysis methods. Furthermore, beyond code construction techniques and

[1] Here, it is important to mention that the importance of Viterbi's work [Vit67] in making the convolutional codes practical cannot be overstated.

theoretical analysis, we also study the VLSI implementation of decoders for LDPC convolutional codes, such that the performance/complexity tradeoff can be examined in its fundamentals.

This dissertation has the following structure. In Chapter 2, we define the LDPC convolutional codes and their encoding and decoding algorithms. We also define the tail-biting versions of the LDPC convolutional codes, which are obtained by applying an wrapping procedure. The encoding and decoding algorithms for the tail-biting codes are also presented.

Chapter 3 presents some structured constructions for LDPC convolutional codes and their tail-biting versions. We start this chapter by presenting some algebraic code construction methods and bounds to their performance parameters. The chapter proceeds with the study of graph configurations that lead to loss of performance. As a remedy, we propose some techniques to avoid such graph configurations during the code construction. Finally, we study the construction of codes based on performance-optimized graph templates. For this purpose, two new construction methods are introduced.

In Chapter 4, we propose a framework for the asymptotic analysis of the performance parameters for LDPC convolutional codes and their tail-biting versions that are derived from performance-optimized graph templates. In the asymptotic regime, we also discuss the relation between tail-biting LDPC convolutional codes and their mother convolutional codes.

A new class of codes called *braided protograph convolutional codes* is presented and analyzed in Chapter 5. Depending on some constraints imposed on the structure of the braided convolutional codes, they can be decoded as LDPC codes or as turbo-like codes. As we will discuss in Chapter 5, the choice between LDPC-like or turbo-like decoding can be done based on complexity considerations.

Chapter 6 presents a new multiple access technique that is based on *orthogonal frequency division multiplexing (OFDM)* and braided convolutional codes [ZLZC05], which we call *braided code division multiple access (BCDMA)*. The BCDMA scheme is our proposal of a channel coding scheme for the multi-user channel.

Chapter 7 discusses the implementation of decoders for LDPC convolutional codes and their tail-biting versions. We start this chapter by algebraically describing the types of parallelism that can be exploited in the implementation of decoders, then we propose the VLSI architectures for the decoders and finish the chapter with synthesis results and a chip implementation, which is capable of achieving very high throughputs.

Finally, conclusions and recommendations for further research are presented in Chapter 8.

We assume that the reader of this dissertation is familiar with the basic concepts of modern algebra and coding theory such as rings, finite fields, isomorphisms, automorphisms, block codes, convolutional codes, *additive white Gaussian noise (AWGN) channel*, *binary erasure channel (BEC)*, minimum distance, free distance, trellis based decoding algorithms, etc. Moreover, some basic knowledge of LDPC codes is also required, such as bipartite (Tanner) graph representations, degree distributions, belief propagation decoding, etc. For the basic concepts of modern algebra and coding theory, we recommend the books by McEliece [McE87], Birkhoff and MacLane [BM77], Peterson and Weldon [PW72], Johannesson and Zigangirov [JZ99], and Lin and Costello [LC04]. For the basics of LDPC codes, we indicate Gallager's monograph [Gal63] and the book by Richardson and Urbanke [RU08].