Cognitive Radio on a Reconfigurable MPSoC Platform

Qiwei Zhang

Composition of the Graduation Committee:

Prof. Dr. Ir.	G.J.M.	Smit, University of Twente, faculty of EEMCS
Dr. Ir.	A.B.J.	Kokkeler, University of Twente, faculty of EEMCS
Prof. Dr. Ir.	C.H.	Slump, University of Twente, faculty of EEMCS
Prof. Dr. Ir.	I.	Niemegeers, TU Delft, faculty of EEMCS
Dr. Ir.	M.J.	Bentum, University of Twente, faculty of EEMCS
Dr. Ir.	D.	Grace, University of York, UK
Prof. Dr.	H.	Zhang, Zhejiang University, China
Prof. Dr.	A.J.	Mouthaan, UT, Faculty of EEMCS (chairman and secretary)

AAF
FREEBAND

This research is conducted within the AAF project supported by the Dutch Ministry of Economic affairs.

University of Twente
Enschede - The Netherlands

The Faculty of Electrical Engineering, Mathematics and Computer Science.
P.O. Box 217, 7500 AE Enschede, The Netherlands.

CTIT
Centre for Telematics and Information Technology

Center for Telematics and Information Technology.
P.O. Box 217, 7500 AE Enschede, The Netherlands.

Printed by Gildeprint
ISBN 978-90-365-2797-2
ISSN 1381-3617, CTIT PhD thesis series No 09-137
DOI 10.3990./1.9789036527972

COGNITIVE RADIO ON A RECONFIGURABLE MPSOC PLATFORM

DISSERTATION

to obtain
the doctor's degree at the University of Twente,
on the authority of the rector magnificus,
prof.dr. prof.dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Thursday the 26th of February 2009 at 15:00

by

Qiwei Zhang

born on the 20th of January 1980
in Lanzhou, China

This dissertation is approved by:

Prof. Dr. Ir.	Gerard J.M. Smit	(promotor)
Dr. Ir.	André B.J. Kokkeler	(assistant promotor)

Abstract

Due to the explosive growth of wireless communication, the demands for radio spectrum are rapidly increasing. It is very difficult to accommodate new wireless services under the current spectrum allocation scheme. On the other hand, the allocated spectrum is not efficiently utilized. Cognitive Radio is proposed as a technology to solve the imbalance between spectrum scarcity and spectrum under-utilization. Spectrum utilization can be improved by making it possible for a user who does not have the license for spectrum (secondary user) to access the spectrum which is not occupied by the licensed user (primary user). This secondary user has the awareness of the spectrum and adapts its transmission accordingly on a non-interference basis. This spectrum access and awareness scheme is referred to as Cognitive Radio. The idea is also known as Dynamic Spectrum Access (DSA) or Open Spectrum Access (OSA). Cognitive Radio is seen as the final point of software defined radio (SDR) platform evolution. A fully flexible and efficient software defined radio platform will be the enabling technology for Cognitive Radio. Cognitive Radio imposes a number of requirements on the processing platform such as flexibility, energy efficiency and guaranteed throughput/latency. The trend in the implementation of SDR is moving towards Multiprocessor System-on-Chip (MPSoC) platforms.

The work of this PhD thesis is part of the Ad-hoc Adaptive Freeband (AAF) project. The aim of the AAF project is to design a Cognitive Radio based wireless ad-hoc network for emergency situations. Although the AAF project addresses Cognitive Radio in a holistic fashion from physical layer to networking issues, the work of this thesis mainly focuses on the design of the adaptive physical layer (baseband processing). The physical layer considered in this thesis mainly consists of two parts: transmission and spectrum sensing. A reconfigurable MPSoC platform is used to support the adaptive baseband processing of Cognitive Radio. A coarse-grain reconfigurable processor called the Montium, developed at the University of Twente, is considered in this thesis as a key element of the proposed MPSoC platform.

The Montium targets the 16 bit Digital Signal Processing (DSP) algorithm domain. It offers a combination of flexibility and energy efficiency.

Although MPSoCs offer many advantages, it is a challenging task to map applications onto MPSoCs, especially highly dynamic applications such as Cognitive Radio. There is a gap between the application models used for the specification of such applications and an optimized implementation of the application on an MPSoC. To close the gap, we propose to use a task transaction level (TTL) interface approach both for developing the Cognitive Radio application at system level and for the platform interface between the application and the proposed MPSoC platform. The TTL approach is used throughout the thesis as the system-level design methodology and its advantages are elaborated by mapping adaptive physical layer algorithms for Cognitive Radio onto the MPSoC platform. The TTL model allows verifying the system's functional behavior and provides profile information for complexity analysis.

The physical layer transmission scheme is a primary design choice for Cognitive Radio. It has to offer agility to access the licensed band on a non interference basis and should make best use of available spectrum. Moreover, it has to provide high data rates similar to other modern wireless systems. OFDM is considered as a prime candidate transmission scheme for Cognitive Radio. In the context of Cognitive Radio, subcarriers of an OFDM system can be deactivated to avoid interference to licensed users. This idea is also known as spectrum pooling. In such an OFDM system for Cognitive Radio, different modulation modes can be loaded onto each subcarrier. This technique, also known as adaptive bit loading, enables Cognitive Radio to optimally use the segmented spectrum. Based on the basic OFDM parameter set used for the AAF system, we propose an adaptive system that combines spectrum pooling and adaptive bit loading.

The profile information generated by the TTL model indicates that the Fast Fourier Transform (FFT) and the Inverse Fast Fourier Transform (IFFT) task are the most computationally intensive parts of the OFDM system. However, due to the deactivation of subcarriers, there could be a large number of zero inputs/outputs for the IFFT/FFT. In this case, the normal radix-2 IFFT/FFT will be inefficient due to the wasted operations on zeros. Therefore, this thesis proposes a novel sparse FFT as an efficient option to reduce the system complexity in case a large number of subcarriers are deactivated. The proposed sparse FFT has been mapped onto the targeted reconfigurable platform. The mapping approach starts from the system-level modeling in the TTL framework. With the TTL model we can verify the algorithm and it provides the profile information to make design tradeoffs at an early design stage. Based on the TTL model, a dynamically reconfigurable FFT module is implemented on the Montium. It enables the reconfiguration of the FFT size and the reconfiguration between sparse FFT and radix-2 FFT. The reconfiguration overhead is small and the sparse FFT gives considerable computation savings in case a large number of subcarriers are deactivated.

One of challenges of OFDM based Cognitive Radio is the appearance of sidelobes which may cause potential interference to licensed systems. Several methods were proposed in literature to mitigate the interference such as deactivating more subcarriers adjacent to the licensed system or applying non-rectangular windows. However, none of them gave a satisfactory solution. Therefore, other multicarrier techniques, such as filter bank multicarrier approaches, are expected to be good alternative transmission schemes for Cognitive Radio. In this thesis, an oversampled filter bank multicarrier system is proposed as an alternative. The proposed filter bank multicarrier system can largely reduce sidelobes to reduce the potential interference. However, the computational complexity of the filter bank multicarrier approach is much higher than the OFDM solution. Since the Montium on the proposed platform is targeted for such computationally complex algorithms, the mapping of the proposed filter bank multicarrier system onto the Montium has been analyzed.

The transmission of Cognitive Radio strictly depends on the reliable detection of the primary user through spectrum sensing. As a result, spectrum sensing is an essential part of Cognitive Radio. Spectrum sensing should also be considered as a part of the physical layer. The major task of the physical layer spectrum sensing is to detect the licensed signal by employing various signal processing techniques. This thesis reviews different signal processing schemes for sensing and focuses on so-called energy detection. An energy based multi-resolution spectrum sensing scheme is proposed in this thesis. The sparse FFT proposed for OFDM based Cognitive Radio also suits this multi-resolution sensing scheme quite well. The filter bank spectrum sensing technique is also considered due to its easy integration with a filter bank multicarrier system.

Samenvatting

Vanwege de explosieve toename van draadloze communicatie neemt de vraag naar radio spectrum toe. Het is bijzonder moeilijk om nieuwe draadloze diensten een plek te geven in de huidige frequentieruimte. Echter, toegewezen spectrum wordt niet op een efficiënte manier benut. Cognitieve Radio wordt beschouwd als een technologie waarmee de onbalans tussen spectrumschaarste en slechte benutting van spectrum kan worden opgeheven. Spectrumgebruik kan worden verbeterd door een gebruiker die geen licentie heeft voor het gebruik van een stuk spectrum (een secundaire gebruiker), toestemming te geven spectrum te benutten dat niet bezet is door de gebruiker die in het bezit is van de licentie (primaire gebruiker). De secundaire gebruiker kent het spectrum en zendt alleen als geen storing wordt veroorzaakt. Het gebruikmaken en bewust zijn van het spectrum wordt aangeduid als 'Cognitieve Radio'. Andere termen die gebruikt worden zijn 'Dynamic Spectrum Access' (DSA) of 'Open Spectrum Access' (OSA). Cognitieve Radio wordt beschouwd als eindstation van de 'Software Defined Radio' (SDR) evolutie. Een volledig flexibel en efficiënt softwaregestuurd radio platform maakt Cognitieve Radio mogelijk. De eisen die Cognitieve Radio aan het platform stelt hebben betrekking op flexibiliteit, energieverbruik en gegarandeerde doorvoersnelheid/vertraging. De trend bij de implementatie van SDR verschuift richting 'Multiprocessor System-on-Chip' (MPSoC) platformen.

Het werk in dit proefschrift vormt een onderdeel van het Ad-hoc Adaptive Freeband (AAF) project. Het doel van het AAF project is om een draadloos ad-hoc netwerk te ontwerpen, gebaseerd op Cognitieve Radio en te gebruiken bij calamiteiten. Hoewel het AAF project Cognitieve Radio benadert door middel van een holistische aanpak, zich uitstrekkend van de fysieke laag tot netwerkaspecten, richt het werk beschreven in dit proefschrift zich voornamelijk op het ontwerp van de adaptieve fysieke laag (basisband bewerkingen). De fysieke laag zoals behandeld in dit proefschrift bestaat uit twee delen: transmissie en 'spectrum sensing'. Om de adaptieve basisbandberekeningen van Cognitieve Radio te ondersteunen worden MPSoC platfor-

men gebruikt. Aan de Universiteit Twente is een grofkorrelig reconfigureerbare processor, Montium genaamd, ontwikkeld en deze wordt beschouwd als een onmisbaar element van het voorgestelde MPSoC platform. De Montium is gericht op het domein van de 16-bit digitale signaalverwerkingsalgoritmen, ook wel ‘DSP’ genaamd. Het biedt een combinatie van flexibiliteit en energie-efficiëntie.

Hoewel MPSoC’s grote voordelen bieden stelt het afbeelden van toepassingen op MPSoC’s, in het bijzonder een uitermate dynamische toepassing als Cognitieve Radio, ons voor grote uitdagingen. Er bestaat een kloof tussen de beschrijvingsmodellen, gebruikt om zulke toepassingen te specificeren, en een geoptimaliseerde implementatie van de toepassing op een MPSoC. Om de kloof te overbruggen stellen we een ‘task transaction level’ (TTL) benadering voor, zowel te gebruiken voor de ontwikkeling van de Cognitieve Radio toepassing op systeemniveau alsook te dienen als platform interface tussen de toepassing en het voorgestelde MPSoC platform. De TTL benadering zal in dit proefschrift toegepast worden als ontwerpmethodologie op systeemniveau en de voordelen zullen worden belicht door het afbeelden van de adaptieve fysieke laag algoritmen voor Cognitieve Radio op het MPSoC platform. Het TTL model maakt het mogelijk het functionele gedrag van het systeem te verifiëren en verschaft kengetallen die gebruikt kunnen worden voor een complexiteitsanalyse.

Welk transmissiemechanisme te gebruiken voor de fysieke laag vormt een eerste ontwerpkeuze voor Cognitieve Radio. Het moet voldoende beweeglijkheid bieden om een secundaire gebruiker, zonder storing te veroorzaken, toegang te bieden tot een band waarvoor een licentie is afgegeven en het moet op een optimale manier gebruik maken van beschikbaar spectrum. Verder moet het voorzien in hoge datasnelheden, vergelijkbaar met andere moderne draadloze communicatiesystemen. OFDM wordt beschouwd als belangrijkste optie voor Cognitieve Radio. In de context van Cognitieve Radio kunnen draaggolven, gebruikt binnen een OFDM systeem, worden uitgeschakeld om storing voor de primaire gebruiker te voorkomen. Dit idee staat bekend als ‘spectrum pooling’. In een dergelijk OFDM systeem voor Cognitieve Radio kunnen verschillende draaggolven op verschillende manieren gemoduleerd worden. Deze techniek, bekend als ‘adaptive bit loading’, maakt het voor Cognitieve Radio mogelijk om op een zo optimaal mogelijke manier gebruik te maken van een gesegmenteerd spectrum. Gebaseerd op de standaard OFDM parameters van het AAF systeem presenteren we een adaptief systeem dat spectrum pooling combineert met adaptive bit loading.

De kengetallen die zijn gegenereerd door middel van het TTL model geven aan dat de ‘Fast Fourier Transform’ (FFT) en de ‘Inverse Fast Fourier

Transform' (IFFT) taken de meeste rekenkracht vergen binnen het OFDM systeem. Echter, doordat draaggolven gedeactiveerd kunnen worden bestaat de kans dat een groot aantal ingangs- en uitgangssignalen van de IFFT/FFT de waarde nul zullen hebben. In dit geval zal de veelgebruikte radix-2 IFFT/FFT inefficiënt zijn vanwege de onnodig uitgevoerde bewerkingen waarvan het resultaat op voorhand vastligt. In dit proefschrift introduceren we daarom een nieuwe 'sparse' FFT die dient als een efficiënt alternatief om de benodigde rekenkracht te verminderen in het geval dat veel draaggolven uitgeschakeld zijn. De sparse FFT is afgebeeld op het reconfigureerbare platform. Het afbeelden begint met de modelering op systeemniveau met behulp van het TTL raamwerk. Met het TTL model verifiëren we het algoritme en het model verstrekt kengetallen die worden gebruikt om afwegingen te maken in een vroeg stadium van het ontwerptraject. Gebaseerd op het TTL model is een dynamisch reconfigureerbare FFT module geïmplementeerd op de Montium. Het is mogelijk de FFT grootte aan te passen en te reconfigureren tussen een sparse FFT en een radix-2 FFT. De inspanning ten behoeve van reconfiguratie is klein en de sparse FFT zorgt voor een aanzienlijke besparing in rekenkracht in het geval dat een groot aantal draaggolven niet gebruikt wordt.

Eén van de uitdagingen van Cognitieve Radio gebaseerd op OFDM is het optreden van zijlussen die mogelijkerwijs storing kunnen vooroorzaken voor primaire gebruikers. In de literatuur worden meerdere methoden voorgesteld om de storing tegen te gaan zoals het deactiveren van meerdere draaggolven die zich dicht bij een frequentie bevinden waarvoor een licentie is afgegeven, of het gebruik van niet-rechthoekige weging. Echter, geen van de oplossingen geeft bevredigende resultaten. Daarom is het de verwachting dat andere technieken die gebruik maken van meerdere draaggolven, zoals een 'filter bank multicarrier' benadering, goede alternatieve transmissiemechanismen kunnen zijn. In dit proefschrift wordt een 'oversampled filter bank multicarrier' systeem voorgedragen als alternatief. Dit mechanisme kan de zijlusniveaus in sterke mate terugdringen om daarmee mogelijke storing te verhinderen. De vereiste rekenkracht van de filter bank multicarrier benadering is veel groter dan voor de OFDM oplossing. Omdat de Montium op het platform dat wij voorstellen juist bedoeld is voor rekenintensieve taken hebben we de afbeelding van het filter bank multicarrier systeem op de Montium geanalyseerd.

Het gebruik van Cognitieve Radio is sterk afhankelijk van een betrouwbare detectie van de primaire gebruiker door middel van spectrum sensing. Spectrum sensing is daarmee een essentieel onderdeel van Cognitieve Radio. Spectrum sensing kan worden beschouwd als een onderdeel van de fysieke laag. De belangrijkste taak van spectrum sensing is het detecteren van signalen van primaire gebruikers, gebruikmakend van verschillende signaalverwerkingstechnieken. In dit proefschrift worden meerdere signaalverwerkingsmechanismen voor sensing beoordeeld en onze speciale aandacht gaat uit naar het zogenoemde 'energy detection'. Een 'multi-resolution' spectrum sensing mechanisme gebaseerd op energy detection wordt in dit proefschrift voorgesteld. De sparse FFT, geïntroduceerd voor Cognitieve Radio gebaseerd op OFDM, is ook uitermate geschikt voor multi-resolution sensing. Een spectrum sensing techniek waarbij gebruik gemaakt wordt van filterbanken is ook bekeken vanwege de gemakkelijke integreerbaarheid met een filter bank multicarrier systeem.

Acknowledgements

Spending four years doing a PhD in a foreign country seems to be a though way of life. However, my journey as a PhD is just the opposite: an exciting, enjoyable and memorable adventure. This adventure was not only about exciting scientific discoveries, but an extraordinary experience for me to see other parts of the world and meet many wonderful people.

Before mentioning all those people who gave me help and support during this memorable journey, I would like to thank my parents, Zhang Ruiquan and Fan Yuxiang. I owe everything to them for what I have achieved so far. Like many other one-child parents in China, they focus their love, attention and ambition on their only child. They have always put education as a priority and encourage me to achieve higher goals in my life. Therefore, becoming a PhD has always been my dream since childhood. Not only did my parents help me set ambitious goals, but they have also been teaching me moral lessons and life philosophies. Although facing great expectations, I have never felt pressure from my parents. Instead, their unwavering support makes me feel more confident in myself. Today, they should be so proud of their son's accomplishments. Therefore, I dedicate this work to my parents for their love and support.

I would like to express my deepest gratitude and respect to my promotor and daily supervisor Prof. Gerard Smit. Gerard is always ready to help me and give me good advice. His optimism and positive attitudes always encouraged me when I was struggling with difficulties in my work. He also gave me a large degree of freedom in research so that I was able to develop my own ideas. My papers and this final thesis benefited a lot from his careful review.

I am so fortunate to have Dr. André Kokkeler as my co-supervisor. My work on Cognitive Radio benefited a lot from numerous discussions with him. His guidance and helpful suggestions made a key contribution to my research. He is always a thorough reviewer who gave critical but helpful comments to my papers and this final thesis. I would also like to thank

André for writing the Dutch abstract for my thesis.

My research was conducted in the AAF project in close collaborations with colleagues within our university and from TU Delft. Marnix Heskamp and Fokke Hoeksma from the Signals and Systems group at the University of Twente provided many useful inputs to my work. Discussions with Dr. Homayoun Nikookar and Ibrahim Budiarjo from the TU Delft have always been a source of inspiration to my research. I also cherish those conference trips made together with Marnix and Ibrahim. I would like to thank Przemyslaw Pawelczak for organizing our demonstrations at the DySPAN conference. Thanks also to Karel Walters and Marcel Hamer, whose MSc projects have contributed to the work in this thesis.

During these years, I have worked with a group of nice colleagues. They have created a friendly and helpful environment. I would like to thank all members and ex-members of the Embedded Systems group.

Many thanks to all members of my graduation committee: Prof. Kees Slump, Prof. Ignas Niemegeers, Dr. Mark Bentum, Dr. David Grace and Prof. Honggang Zhang.

Special thanks to Prof. Pieter Hartel, who helped me to start my PhD at the University of Twente.

I am very grateful to my MSc supervisor Prof. Lajos Hanzo in the University of Southampton. Those lessons and advice I learned from him are extremely helpful to my research. Probably Prof. Hanzo is the most knowledgeable man whom I have ever met in the field of wireless communication. As a great scholar, he has always been my example.

During this four years' stay in Enschede, I have met many Chinese and international friends who made my life more colorful. I would like to mention some of them in particular. Can you imagine that I could meet a Chinese who has the exactly same date of birth with me in Holland? Even more interestingly, we have the same family name and work in the same building. People might even start to think we are twins. Maybe because of all these coincidences, I feel a special intimacy with Zhang Yang who has been my best friend in Enschede. I am always grateful to Feng Xiaozhou for her help and care during my Achilles injury, which was the darkest memory during my PhD. Thanks to Wu Nan for those travels we have done together. Thanks to Ma Bin for being a pleasant flat mate. Thanks to Li Rongmei for nice chatting and delicious food. Thanks to Guo Yuanqing for her useful tips for doing a PhD. Michel Rosien, a former colleague of mine, has been my best Dutch friend. It was Michel who drove me to the hospital when I had my Achilles injury. I also enjoyed those chess (both Chinese and international) games with him. Thanks to Ivan Lakhturov for being an affable and interesting flat mate. In those days living in the international dormitory, I met people from every corner of the world. It was an incredible experience for me to learn different culture and of course taste different food from all over the world. I would like to thank Yosie, Cem, Magi and all other international friends in Witbreuksweg 379 for this incredible memory. I also want to thank all my football friends for those enjoyable games, although I don't even know some of their names.

I would like to thank all my relatives and friends back in China for their support and for being in touch with me so that I never felt home sick. Of course I shouldn't forget to thank all those friends I have met on the internet.

Once again, thank you all for being part of the greatest memory in my life!

Enschede, January 2009
Qiwei Zhang

List of Figures

List of Tables

Table of Contents

Chapter 1

Introduction

Large parts of the assigned radio spectrum is underutilized while the increasing number of wireless multimedia applications leads to spectrum scarcity. Cognitive Radio is proposed as a promising technology to utilize non-used parts of the spectrum that actually are assigned to primary services. This chapter[1] *introduces the background of Cognitive Radio and outlines the work described in this thesis in the context of Cognitive Radio. The work described in this thesis is a part of the Adaptive Ad-hoc Freeband (AAF) project. A short description of the AAF project is also given.*

[1] Parts of this chapter have been published in publication [QZ3] and [QZ6].

1.1 Motivation and Background

Due to the explosive growth of wireless communication, the demand for radio spectrum is rapidly increasing. Under the current spectrum allocation scheme, each new wireless system will be assigned a fixed frequency band. However, most of the spectrum has already been assigned to existing wireless systems. As a result, it is rather difficult to accommodate new wireless services under the current scheme. Therefore, it is imperative to come up with a new spectrum access and allocation scheme. Although much of the spectrum has been allocated, the actual usage of the assigned spectrum is rather sparse in terms of time, frequency and location. Figure 1.1 shows a typical spectrum utilization of the 400-800MHz frequency band during 10 minutes time span in Twente, The Netherlands. The color indicates the energy level on the frequency band. The darker, the higher the level of the energy emission is. Clearly we can see plenty of underutilized spectrum (white space), where no energy is emitted most of the time. In November 2002, the Federal Communications Commission (FCC) in the United States released a report [27] aimed at improving the management of spectrum resources in the US. The report concluded that the current spectrum scarcity problem is largely due to the strict regulation on spectrum access. Spectrum utilization can be improved by making it possible for a *secondary user* (who does not have the license for that spectrum) to access the spectrum which is not occupied by the licensed user (*primary user*). This secondary user needs

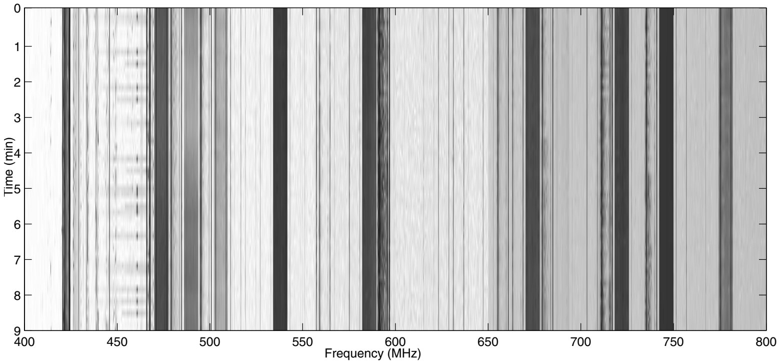

Figure 1.1: The spectrum utilization in 10 minutes (400MHz-800MHz) in Twente, The Netherlands

to have awareness of the spectrum and adapts its transmission accordingly on a non-interference basis. This spectrum access and awareness scheme is referred to as Cognitive Radio by the FCC. The idea is also known as Dynamic Spectrum Access (DSA) or Open Spectrum Access (OSA).

The original idea of Cognitive Radio was proposed by Joseph Mitola in his paper [57], where he proposed that Cognitive Radio can enhance personal wireless services by a Radio Knowledge Representation Language (RKRL). This language represents knowledge of all aspects of radio, from transmission to application scenarios, in such a way that automated reasoning about the needs of the user is supported. Cognitive Radio is able to autonomously observe and learn the radio environment, generate plans and even correct mistakes. A comprehensive conceptual architecture of Cognitive Radio was later presented in his PhD thesis [58], where Cognitive Radio was thought as a final point of the software-defined radio platform evolution: a fully reconfigurable radio that changes its communication functions depending on network and/or user demands. Mitola's work covered interesting research subjects in multiple disciplines such as wireless communication, computer science and cognitive science. His original work opened a new research area and still stimulates researchers to come up with new ideas for Cognitive Radio.

The FCC focuses on the dynamic spectrum access aspect of the original concept brought up by Mitola. This focus has also become a theme of recent research on Cognitive Radio, including the AAF project[2]. The impact of this research may fundamentally change the current status of radio communication.

1.1.1 State-of-the-Art

Cognitive Radio attracted a lot of interests around the world since it first appeared and especially since the recent focus on dynamic spectrum access. We mention a few research projects related to Cognitive Radio around the world.

The Berkeley Wireless Research Center (BWRC) has a dedicated research project on Cognitive Radio[3]. A white paper [6] on Cognitive Radio defined the scope of their work. Their focus is on dynamic spectrum Cognitive Radio, as it was mentioned in the FCC initiative. They treat the subject in a holistic fashion: from physical layer issues [11] to MAC layer

[2] AAF project website: `http://aaf.freeband.nl`

[3] Berkeley Wireless Research Center Cognitive Radio Research project website: `http://bwrc.eecs.berkeley.edu/Research/Cognitive/home.htm`

issues and from analog frontend to the computing platform supporting baseband processing [55]. It is interesting to look into their study on spectrum sensing. From the system perspective, they study some basic considerations (see [56] and [92]): the link budget of the sensing; the effect of noise on sensing; the cooperation of sensing nodes. From the signal processing perspective, a comparison study is made on different sensing techniques. Some implementation considerations related to sensing are also presented [10].

Spectrum pooling [103] is investigated by Weiss from the University of Karlsruhe. The basic idea is that a secondary user can dynamically access the licensed band by switching on and off OFDM subcarriers to avoid interference to the licensed user (primary user). However, the spectrum power leakage in the FFT based traditional OFDM systems could cause potential interference to the licensed system. They also observed this in [104], where two counter measures are proposed: spectrum shaping and switching off subcarriers adjacent to the licensed user (primary user). There are other challenges for spectrum pooling such as detection of spectrum access and synchronization, see [103].

The Cognitive Radio project[4]at Virginia Tech does not only aim to improve spectrum utilization but they treat the radio as a biological system. In [72], a genetic algorithm based cognitive engine is proposed to learn its environment and respond with an optimal adaption. This approach to Cognitive Radio is more or less similar to the original concept of Mitola. They also proposed to apply the Cognitive Radio concept to public safety networks [49].

The key objective of the European Union 6th framework End-to-End Reconfigurability (E2R) project[5]is to devise, develop and trial the architectural design of reconfigurable devices and supporting system functions. Different views are addressed, ranging from users, application and service providers, operators, and regulators in the context of heterogeneous systems. Although the project does not specifically address Cognitive Radio, dynamic spectrum allocation and evolution from software defined radio to Cognitive Radio has been envisioned.

The KDDI R&D Laboratories[6]in Japan proposed a Cognitive Radio scheme by using multiple transmission links [91]. A virtual link between wireless stations for Cognitive Radio transmission is formed by bundling multiple physical wireless links. The system enables to assign packets to

[4] Virginia Tech. Cognitive Radio research website:
`http://www.cognitiveradio.wireless.vt.edu/`

[5] E2R project website: `http://e2r.motlabs.com`

[6] KDDI website: `http://www.kddilabs.jp`

the optimum wireless link and to avoid congestion in a specific frequency band. The system can use existing air interfaces without designing a new physical layer. A similar idea has been employed by the FIGO system developed by the TI-WMC[7]. The FIGO system can use multiple radios creating a high-capacity and low latency network with small delays. Further, it can pro-actively optimize routing and opportunistically map traffic to radio conditions and to the network topology.

In parallel with the ongoing research projects around the world, international standardization organizations also have made proposals to improve the spectrum utilization. An example is the IEEE 802.22 standard[8], which is a new standard for a cognitive point-to-point (P2P) air interface for spectrum sharing within television bands. Television channels are very suitable for cognitive radio because they have a relatively unique spectrum signature that is easy for a cognitive radio to identify. The signals are also rigidly assigned to 6-MHz-wide channels with fixed center frequencies.

The Defense Advanced Research Projects Agency (DARPA) in the US is also interested in the application of Cognitive Radio to future military communications. The Next Generation (XG) program[9]is their effort to develop both the enabling technologies and system concepts to dynamically redistribute allocated spectrum along with novel waveforms in order to provide improvements in assured military communications. They have successfully demonstrated a prototyped Cognitive Radio system in a field test [75]. In the test, XG radio nodes are able to identify unused spectrum and establish a communication network with little interference to existing legacy systems. The results [54] show the feasibility of dynamic spectrum Cognitive Radio with existing technology. This test has been one of the most successful Cognitive Radio demonstrations being published.

1.1.2 The AAF project

The aim of our Ad-hoc Adaptive Freeband (AAF) project is to design a Cognitive Radio based wireless ad-hoc network for emergency situations. Current day emergency services rely for data communications on public radio networks like GPRS. Sometimes, e.g. in disaster situations, even GSM is used for voice communication between relief workers. However, in case of emergency the public networks may get overloaded. Moreover, the relief network must be able to handle multimedia signals and has to deal

[7] Twente Institute for Wireless Mobile Communications website: `www.ti-wmc.nl`

[8] IEEE 802.22 Standard: `http://www.ieee802.org/22`

[9] XG radio website: `http://www.darpa.mil/sto/smallunitops/xg.html`

with large, possibly unpredictable amounts of data. The generally available public networks are not considered to be reliable enough for emergency situations because public networks lack capacity (e.g. in offered data rates or multimedia traffic support) and are susceptible to the destruction of their infrastructures. If dedicated emergency networks (e.g. C2000[10]) are used, a major drawback is their spectrum scarcity because current emergency networks are assigned with a limited spectrum and fixed bandwith [63]. The large amounts of multimedia data in the emergency networks during disasters require a large amount of radio resources. One band can easily get congested due to heavy traffic, which makes it inadequate for emergency use. If several fragmented bands are assigned to emergency use, the interoperability and the lack of standards will become another problem [20]. Therefore to alleviate this spectrum shortage problem, a radio which dynamically accesses free spectrum resources turns out to be an interesting solution.

Although the AAF project addresses Cognitive Radio in a holistic fashion from physical layer to networking issues, the work described in this thesis mainly focuses on the physical layer related issues including baseband transmission, spectrum sensing and a reconfigurable platform to support physical layer processing. The physical layer is a very important part of any communication system including Cognitive Radio. The design of the physical layer has a profound impact on the feasibility of communication processes at higher layers [99]. In the AAF project three issues are dealt with in more detail:

- Spectrum sensing is a process which identifies the free spectrum resources. On the physical layer level, intensive signal processing has to be done to obtain an estimation of the spectrum and detect licensed user signals. Reliable sensing is the first step to generate the spectrum occupancy information based on which the Cognitive Radio nodes in the network will establish communications.

- Multicarrier transmission has been considered for Cognitive Radio since it offers opportunities to optimally use the segmented spectrum. In the AAF project, the baseband transmission system borrows the idea of spectrum pooling [103], switching off subcarriers to avoid potential interference to licensed users and to optimally use the remaining spectrum.

[10] C2000 network homepage (in Dutch): `http://www.c2000.nl`

- Cognitive Radio has to operate in different frequency bands, combat various negative effects of wireless channels and support various multimedia services. Therefore Cognitive Radio needs physical layer adaptivity. This adaptivity has to be supported by a software defined radio platform. Such a platform has to provide sufficient flexibility which means the same hardware has to be reconfigured to perform different tasks. At the same time, the platform has to guarantee the performance of high data rate multimedia communications. Moreover, the power consumption is a major concern for battery powered mobile radio devices. Therefore, the software defined radio platform considered for Cognitive Radio has to offer an optimal combination of reconfigurability, performance and energy efficiency. Only with such a platform, Cognitive Radio can be brought from a novel idea to reality.

An overview of the physical layer and platform architecture of a Cognitive Radio node in the AAF network is shown in figure 1.2.

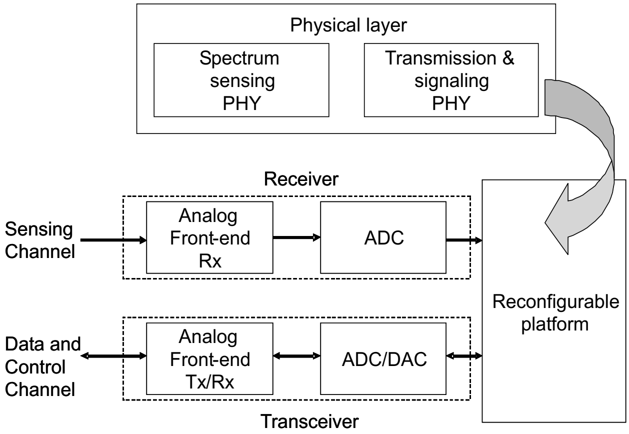

Figure 1.2: The physical layer and platform architecture of a Cognitive Radio node

Within a Cognitive Radio node, three types of functional channels are supported. Two of these functional channels, the data channel and control channel are multiplexed onto the same transceiver in figure 1.2.

- **A Sensing Channel** is dedicated to spectrum sensing. It constantly listens to the radio environment and searches for unoccupied spectrum.

- **A Data Channel** is used to exchange the user data. Multicarrier transmission is proposed in our project.

- **A Control Channel** is needed for the control information exchange between Cognitive Radio nodes. This information includes frequency occupancy information and control information. In the scope of our work, we assume such a control channel already exists. This assumption will be used throughout the thesis.

The physical layer of a Cognitive Radio node considered in this thesis includes the data transmission and signaling physical layer and the spectrum sensing physical layer. The data transmission and signaling physical layer is common for all wireless communication systems. However, it has to be specially tailored for Cognitive Radio which is fundamentally different from the traditional wireless communication systems in the way it accesses the spectrum. As a unique feature of Cognitive Radio to search for unused spectrum, spectrum sensing should be included as an essential part of the physical layer.

In the proposed platform architecture, a Cognitive Radio node consists of a dedicated receiver for spectrum sensing which is independent of the data transmission. A transceiver is used for the exchange of data and control information. The transceiver and sensing receiver are connected to the antenna(s) via an analog front-end that targets the band of interest. We are aware that receiver and transceiver schemes are not limited to the scheme proposed in this thesis. For example, a dedicated transceiver may be needed for control information if the transmission schemes of control information and data are different. Moreover, spectrum sensing can share the same transceiver hardware with data and control information transmission in a time division fashion. Although the receiver and transceiver schemes may vary, the baseband processing of the Cognitive Radio physical layer is all done on a single reconfigurable platform. Cognitive Radio has a list of requirements for the processing platform:

- **Flexibility**

  To support the adaptive physical layer of Cognitive Radio, the platform has to be reconfigurable to be able to perform different tasks/algorithms.

- **Energy Efficiency**

  Energy-efficiency is an important design issue for battery powered portable devices such as Cognitive Radio. The functionality of these devices is strictly limited by the energy consumption. There is an exponential increase in demand for streaming communication and processing for wireless baseband processing and multimedia applications. Such demand is even higher for Cognitive Radio which has to constantly monitor the spectrum and adapt its transmission accordingly. However, the energy content of batteries is only increasing at a pace of 10% per year.

- **Guaranteed Throughput/Latency**

  The processing platform has to guarantee throughput and latency to provide the required Quality-of-Service (QoS) to the user.

- **Predictability and Composability**

  Due to its adaptivity, Cognitive Radio systems are very complex. To manage such complexity, predictable techniques are needed. Another reason for using predictable techniques is composability which means that in case multiple applications are mapped on the same platform, the behavior of one application should not influence another application.

- **Design Flow**

  Design automation tools form the bridge between processing hardware and application software. Design tools are a critical requirement for the viability of the platform. Cognitive Radio may evolve quickly to include updated features. Such tools help to reduce the design cycle (i.e. lower costs and shorter time-to-market).

To summarize, mapping of the adaptive baseband processing of the Cognitive Radio physical layer on the reconfigurable platform is the heart of our study and the focus of this PhD thesis.

1.2 Research Objectives

The main objective of this research is to design, validate and implement high performance, adaptive and efficient physical layer digital signal processing (DSP) algorithms of Cognitive Radio onto a reconfigurable platform.

To reach this main objective, several sub-objectives have been established in this thesis as follows:

- The choice of the reconfigurable platform is an important design decision. Based on the list of requirements in section 1.1.2, various hardware architectures, for software defined radio should be compared to make the choice.

- The design of physical layer algorithms for Cognitive Radio poses new challenges but also gives opportunities for innovations. DSP algorithms for the Cognitive Radio physical layer can be further divided into two categories based on functionality: baseband transmission and spectrum sensing.

  For the baseband transmission, the thesis focuses on the spectrum pooling type of systems [103] and identifies the following research issues:

  – Multicarrier transmission is considered for spectrum pooling based Cognitive Radio. In spectrum pooling, Cognitive Radio has to deactivate a number of subcarriers to avoid interference to the licensed user. Deactivated subcarriers result in a special signal structure which gives opportunities to develop new algorithms to achieve more efficient computation.

  – In spectrum pooling, mutual interference and interference from Cognitive Radio to the licensed system in particular are major challenges for successful coexistence. The physical layer design should mitigate such interference to a large extent.

  For designing physical layer spectrum sensing algorithms, the following issues should be considered:

  – The performance of the licensed signal detection should be acceptable to satisfy certain detection probabilities.

  – The algorithm should have a low computational complexity to reduce the sensing time. For example, Cognitive Radio should adapt its transmission quickly enough to avoid interference to an emerging licensed user.

  – The algorithm should be reusable for both spectrum sensing and baseband transmission.

- Mapping the Cognitive Radio physical layer onto the reconfigurable platform is our final goal. The design methodology should have the following features:

  - It should follow a top-down approach and give a system-level description of the application.
  - It should close the gap between the algorithm development and the actual implementation on the platform.
  - It should provide performance analysis which in turn shapes algorithm development.

These objectives are not separated or independent from each other. Instead, they are mutually dependent and should be treated as a whole. For example, the choice of the platform depends on what kind of algorithms will be supported while the algorithm design should consider the constraints of the platform. The design methodology should be suitable for the targeted hardware platform and the application to be mapped. Moreover, the design methodology can shape the algorithm design by providing the profile information of the mapping. Therefore, in the subsequent chapters we will treat these topics in an interwoven fashion.

1.3 Thesis Contributions

This thesis presents the following novel contributions in the areas of Cognitive Radio, embedded system design and signal processing:

- We propose to use a reconfigurable MPSoC platform to support the adaptive baseband processing of Cognitive Radio. Cognitive Radio is seen as the final point of software defined radio platform evolution. The trend in the implementation of SDR is moving towards Multiprocessor System-on-Chip (MPSoC) platforms to fulfill the requirements such as flexibility, energy efficiency, guaranteed throughput and latency. Cognitive Radio on a reconfigurable MPSoC is future oriented. Reconfigurable architectures and coarse grained reconfigurable architectures in particular will be key elements in such MPSoC platforms. The Montium architecture developed in our group is considered as the targeted coarse grained reconfigurable architecture in this thesis.

- We propose to use the task transaction level (TTL) interface approach both for developing the Cognitive Radio application at system level and for the platform interface between the application and the proposed MPSoC platform. The TTL model allows verifying the functional behavior of the system and provides profile information for complexity analysis.

- We present an adaptive system that combines spectrum pooling and adaptive bit loading as the transmission scheme for Cognitive Radio. In the context of Cognitive Radio, subcarriers of an OFDM system can be deactivated to avoid interference to licensed users. This idea is also known as spectrum pooling. In such an OFDM system for Cognitive Radio, different modulation modes can be loaded onto each subcarrier. This technique, also known as adaptive bit loading, enables Cognitive Radio to optimally use the segmented spectrum. An adaptive OFDM system based on the basic OFDM parameter set used for the AAF system is modelled in TTL. The profile information from TTL confirms that the FFT/IFFT task is the most computationally intensive task in OFDM.

- We propose a novel sparse FFT for OFDM based Cognitive Radio. Due to the deactivation of subcarriers, there could be a large number of zero inputs/outputs for the IFFT/FFT in an OFDM based Cognitive Radio system. In this case, the normal radix-2 IFFT/FFT will be inefficient due to the wasted operations on zeros. The proposed sparse FFT is an efficient option to reduce the system complexity in case a large number of subcarriers are deactivated. The proposed sparse FFT has been mapped onto the targeted reconfigurable platform. The mapping approach starts from the system-level modeling in the TTL framework. With the TTL model we can verify the algorithm and it provides the profile information to make design tradeoffs at an early design stage. Based on the TTL model, a dynamically reconfigurable FFT module is implemented on the Montium. It enables the reconfiguration of the FFT size and the reconfiguration between sparse FFT and radix-2 FFT. The reconfiguration overhead is small and the sparse FFT gives considerable computation savings in case a large number of subcarriers are deactivated.

- We propose an oversampled filter bank multicarrier system as an alternative transmission scheme for Cognitive Radio. One of challenges of OFDM based Cognitive Radio is the appearance of sidelobes which may cause potential interference to the licensed system. The proposed filter bank multicarrier system can largely reduce sidelobes to reduce the potential interference and achieve slightly better performance than OFDM. However, the computational complexity of the filter bank multicarrier approach is much higher than the OFDM solution. Since the Montium on the proposed platform is targeted for such computationally complex algorithms, the mapping of the proposed filter bank multicarrier system onto the Montium has been analyzed.

- We propose an energy based multi-resolution spectrum sensing scheme. The sparse FFT proposed for OFDM based Cognitive Radio suits this multi-resolution sensing scheme quite well. The dynamically reconfigurable FFT module for OFDM transmission can be reused for multi-resolution sensing. The filter bank spectrum sensing technique is also considered and the mapping onto the Montium is discussed.

1.4 Thesis Organization

The thesis is organized as follows:

Chapter 2 introduces different hardware architectures and compares their features in the context of supporting the physical layer processing of Cognitive Radio. A heterogenous reconfigurable MPSoC platform for Cognitive Radio is presented. Coarse-grained reconfigurable hardware, the key element in the proposed MPSoC platform is introduced and the emphasis is on the Montium. Finally we introduce the TTL design methodology.

Chapter 3 focuses on OFDM based Cognitive Radio. We start with some fundamentals of OFDM including some example OFDM systems. The design of a baseline OFDM system in the context of AAF is considered. We propose an adaptive OFDM system which combines spectrum pooling and adaptive bit loading. The adaptive OFDM system is modelled at system level with the TTL interface approach.

Chapter 4 proposes a novel sparse FFT for OFDM based Cognitive Radio. We start with the motivation for the sparse FFT, followed by a survey of related work. The algorithm is presented in detail and followed by a complexity analysis. Implementation of a dynamically reconfigurable FFT module on the Montium based platform is presented.

Chapter 5 focuses on a filter bank multicarrier system for Cognitive Radio. As one of contributions of this thesis, an oversampled filter bank multicarrier system is presented. Other filter bank multicarrier systems are reviewed in the context of Cognitive Radio and some challenges for filter bank multicarrier systems are presented. Finally, the mapping of the proposed filter bank multicarrier system onto the Montium is presented.

Chapter 6 discusses various physical layer spectrum sensing schemes for Cognitive Radio. Two categories of sensing schemes have been considered, namely energy detection and feature detection. Energy detection is our focus. We propose a multi-resolution energy detection scheme based on the sparse FFT. In addition, Filter bank spectrum sensing is also considered as an energy detection scheme. Finally, the mapping of various sensing schemes onto the Montium based platform is considered.

Chapter 7 concludes this thesis by outlining the research achievements and future research directions.

Chapter 2

Hardware Platforms and Design Methodology for Cognitive Radio

Cognitive Radio is seen as the final point of Software Defined Radio (SDR) platform evolution. A fully flexible and efficient software defined radio platform will be the enabling technology for Cognitive Radio. This chapter[1] *focuses on the hardware architectures for a software defined radio platform which will eventually evolve to Cognitive Radio as well as the necessary design methodology for the platform. An overview of common embedded computer architectures will be given. A heterogenous reconfigurable multiprocessor System-on-Chip platform is proposed to support Cognitive Radio. The key element on such a platform is coarse-grained reconfigurable hardware. A design methodology is proposed for mapping Cognitive Radio onto the targeted platform.*

[1] Parts of this chapter have been published in publication [QZ5] [QZ10].

2.1 Introduction

As already foreseen by Mitola [58], Cognitive Radio is the final point of Software Defined Radio (SDR) platform evolution: a fully reconfigurable radio that changes its communication functions depending on network and/or user demands. To make this evolution reality, a fully flexible and efficient SDR platform is the most important enabling technology. A SDR platform for Cognitive Radio should be able to support a broad range of frequencies, air-interfaces and various services. Moreover, a SDR platform for Cognitive Radio should be able to dynamically reconfigure its functionality to fulfill the adaptivity requirements of Cognitive Radio. Another concern for Cognitive Radio devices like for other battery-powered wireless devices is energy efficiency. The trend in the implementation of SDR is moving towards Multiprocessor System-on-Chip (MPSoC) platforms which combine flexibility, performance and energy efficiency. This trend implies that future MPSoC based platforms are good candidates to support Cognitive Radio. This chapter is dedicated to hardware architectures for Cognitive Radio. Section 2.2 introduces several common embedded computer architectures for SDR. A heterogenous reconfigurable MPSoC platform for Cognitive Radio is presented in 2.3. Coarse-grained reconfigurable hardware, the key element in the proposed MPSoC platform, is introduced in section 2.4. The emphasis on section 2.4 is on a coarse-grained reconfigurable architecture developed in the CAES group at the University of Twente and a short overview of related work follows. A design methodology is introduced in section 2.5 for mapping Cognitive Radio onto the proposed MPSoC platform. The last section summarizes the chapter.

2.2 Hardware Architectures

Different types of hardware architectures can be found in embedded systems for wireless communications. Several commonly used architectures include: General Purpose Processors (GPPs), Digital Signal Processors (DSPs), reconfigurable hardware and Application Specific Integrated Circuits (ASICs). These hardware architectures have different characteristics regarding flexibility, performance and energy efficiency [39]. Generally speaking, the more flexibility architectures (such as GPPs) have, the less efficient in terms of performance and energy consumption they are. The ASIC solutions are most efficient in terms of performance and energy consumption, however, they do not offer flexibility for different tasks. In between is reconfigurable hardware such as Field Programmable Gate Array (FPGA). It offers tradeoffs between performance and flexibility.

2.2.1 General Purpose Processor

The GPPs are the most commonly used hardware architectures in embedded systems. They can be programmed to perform almost any algorithm. There are basically two kinds of GPP architectures with different memory organizations: the *Von Neumann architecture* and the *Harvard architecture*. The *Von Neumann architecture* has a shared memory for both data and instructions. The memory organization of the *Von Neumann architecture* limits the data throughput between the memory and the processor because data and instructions can not be fetched at the same time. This limitation is also known as the Von Neumann bottleneck. To relieve the Von Neumann bottleneck, an architecture known as the *Harvard architecture* splits the memory into separate data and instruction memory. Another technique to improve memory fetching is introducing caches. A cache is a small and fast memory which stores a subset of the content of a large and slow memory. Nowadays, a GPP with a separate data and instruction cache is also regarded as a *Harvard architecture* although it may have a single main memory for both data and instructions.

The Arithmetic Logic Unit (ALU) of the GPP is designed to support simple but general operations such as *Add* and *Shift*. To perform complex operations, the GPP has to combine several simple operations. An algorithm is a set of operations executed on the processor. With the help of mature tooling, designers only need to develop the software in high level computer languages for a given algorithm and the compiler will generate the operations to be performed on the machine. So, the GPP is very flexible, it can support different algorithms and it is easily programmable. However, there are some drawbacks of the GPP:

- it is designed for general purpose computing, thus may not be efficient for a specific algorithm with complex operations.
- it has to fetch and decode instructions before execution.
- to maintain its performance, it has to run at a high clock frequency which becomes a bottleneck for power optimization.

Therefore, the GPP is not a good candidate for computationally intensive tasks in wireless baseband processing.

2.2.2 Digital Signal Processor

Compared with a GPP, a Digital Signal Processor (DSP) has enhanced features for digital signal processing algorithms. It has dedicated hardware to support typical DSP operations such as Multiply Accumulate (MAC), modulo and bit reverse. Moreover, the DSP hardware architecture has been optimized to exploit Data-Level Parallelism (DLP) and Instruction-Level Parallelism (ILP). However, one major drawback remains for the DSP: fetching and decoding every instruction. This drawback still results in a considerable overhead in power consumption. Therefore, the DSP is sometimes considered as a special type of GPP.

2.2.3 Application Specific Integrated Circuit

Unlike the GPP, an ASIC is dedicated hardware, designed for a specific function or application. An ASIC implementation for a specific algorithm is often optimized for speed, size and energy efficiency. However, the circuitry of an ASIC is fixed after fabrication. Therefore, its function can not be changed to support new applications. The effort of designing an ASIC is rather high and time consuming, thus the time to market could be relatively long.

A large number of ASICs are used for today's wireless standards to achieve optimal performance and energy efficiency, especially in mobile handsets. However, evolving wireless standards result in evolving specifications that have to be supported by hardware. Moreover, future wireless systems such as Cognitive Radio may change its functionality dynamically. Because of their lack of flexibility, ASICs are considered to be unsuitable for future wireless communications, especially for highly flexible Cognitive Radio.

2.2.4 Reconfigurable Hardware

The ASIC is inflexible while the GPP is not efficient in terms of performance and power. None of them offer the combination of flexibility, performance and energy efficiency required for future wireless mobile devices. In this thesis, the applicability of reconfigurable hardware for Cognitive Radio is researched. Reconfigurable hardware contains a reconfigurable fabric with which customer functionalities can be built. Unlike an ASIC, no new fabric needs be designed for each new application. Moreover, the functionalities implemented in the reconfigurable fabric can change over time according to the environment or usage changes in the system. For example, different signal processing algorithms have to be applied in Cognitive Radio in different transmission modes. Furthermore, an optimal computing structure can be produced using a reconfigurable fabric to match the application requirements. Therefore, reconfigurable hardware can be more efficient than a GPP in terms of performance and power. A study [88] reports that moving critical software loops to reconfigurable hardware results in average speedup of 3 to 7 times and energy saving of 35% to 70%.

Reconfigurable devices are often classified as either coarse-grained or fine-grained. The classification is mainly based on the granularity of operations, which defines the size of the smallest functional block that can be configured for a specific operation. Reconfigurable hardware with different levels of granularity offers different degrees of flexibility. A good comparison and classification of different reconfigurable devices can be found in [93].

- **Fine-grained:** A fine-grained reconfigurable device can implement a logic function at bit level. The most common fine-grained devices are Field Programmable Gate Array (FPGA) families. An FPGA typically consists of a matrix of interconnected logic cells. Both logic cells and interconnections can be configured at the bit level to implement a certain function. Fine-grained reconfigurable hardware can achieve a high degree of flexibility but result in significant overhead of area, delay and power consumption. To reduce this overhead, some modern FPGA devices (e.g. Altera, Xilinx) are enhanced by embedding coarse-grained functional units.

- **Coarse-grained:** A coarse-grain reconfigurable device is configured at word level. Functional units such as multipliers and adders are introduced. Although the introduction of coarse functional units reduces flexibility, coarse-grained reconfigurable hardware can be optimized for a particular algorithm domain such as DSP algorithms for wireless communications. For this reason, we also refer to coarse-grained reconfigurable hardware as Domain Specific Reconfigurable Hardware (DSRH). Driven by the market demands for reconfigurable and energy efficient mobile devices, there has been growing interests in coarse-grained reconfigurable hardware [87]. Many groups have investigated and successfully developed coarse-grained reconfigurable systems [93, 87]. The Montium [39] processor developed in our group is one of the successful examples. A detailed introduction on the Montium and related work on coarse-grained reconfigurable architectures will be given in section 2.4.

From the comparison and discussion of different hardware architectures, we may conclude that reconfigurable hardware and coarse-grained reconfigurable hardware in particular will become an important element in future wireless devices (e.g. Cognitive Radio).

2.3 Heterogenous Reconfigurable Multiprocessor System-on-Chip

As originally indicated by the well known Moore's law [59], the transistor density of Integrated Circuit (IC) doubles every 12 months. However, the recent development in semiconductor industry shows that density doubles every 18 months. The increase of transistor density means more computing resources become available in the same silicon area. Therefore, more and more functionalities and even a whole electronic system can be integrated into a single chip. The System-on-Chip (SoC) concept has become popular in the area of embedded systems. Nowadays a SoC often contains multiple, usually heterogenous, processing elements. This technology is known as Multiprocessor System-on-Chip (MPSoC) or multi-core SoC. MPSoC technology is increasingly used in SDR. As predicted in a recent publication [67], MPSoC based SDR cell phones are expected to make an inroad in 2010 and to dominate from 2015 on, although challenges due to the system complexity have to be overcome. MPSoCs are expected to be the future platform to support SDR, which will eventually evolve to Cognitive Radio. Therefore, we propose a heterogeneous reconfigurable MPSoC platform to support adaptive baseband processing of Cognitive Radio [112].

2.3.1 A Template Tile MPSoC Architecture

The heterogeneous reconfigurable MPSoC platform we propose for Cognitive Radio is presented in figure 2.1. It is a tiled MPSoC architecture (or multi-core architecture) template including different interconnected heterogeneous tile processors: fine-grained reconfigurable tiles (e.g. embedded FPGAs), coarse-grained reconfigurable cores (e.g. Domain Specific Reconfigurable Hardware (DSRH)), general purpose programmable cores (e.g. DSPs and microprocessor cores) and memory blocks. All cores are interconnected by a Network-on-Chip (NoC). This NoC consists of many on-chip routers (R in figure 2.1 stands for router). The reason for heterogeneity is that typically, some algorithms run more efficiently on fine-grained reconfigurable architectures (e.g. PN-code generation), some perform optimal on coarse-grained

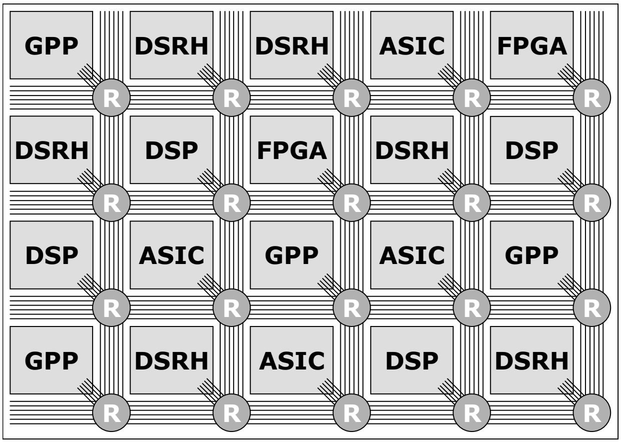

Figure 2.1: Heterogeneous multiprocessor tile SoC

reconfigurable architectures (e.g. Frequency Impulse Response(FIR) filters or Fast Fourier Transform(FFT) algorithms) and control oriented tasks are best processed on general purpose cores [70]. Application designers, high-level compilers or a run-time system can choose the most efficient processing core for the type of processing needed for a given application task.

A multi-core architecture has a number of advantages [82]:

- It is a future-proof architecture, as the processing cores do not grow in complexity with technology. Instead, as technology scales, simply the number of cores on the chip grows.
- A multi-core organization can contribute to the energy-efficiency of a SoC. The best energy savings can be obtained by simply switching off cores that are not used, which also helps for reducing the static power consumption. Furthermore, the locality of reference (the phenomenon that the same value or related storage locations being frequently accessed) principle can be exploited by each core. Moreover, a core processor might not need to run at full clock speed to achieve the required Quality-of-Service (QoS) at a particular moment in time (i.e. Dynamic Voltage and Frequency Scaling per core).
- When one of the cores is discovered to be defect, this defective core can be switched-off and isolated from the rest of the design.
- A multi-core approach also eases verification of an integrated circuit

design, since the design of identical cores only has to be verified once.

- A multi-core architecture can be reconfigured partially and dynamically. In a multi-core architecture every processing core is configured independently. For example, unused cores can be configured for new tasks, while at the same time other cores continue performing their tasks.

A heterogenous multi-core architecture combines performance, flexibility and energy-efficiency. It supports high performance through massive parallelism, it matches the computational model of the algorithm with the granularity and capabilities of the processing entity, it can operate at minimum supply voltage and clock frequency and hence provides energy-efficiency and flexibility at the right granularity only when and where needed and desirable. Dynamic reconfiguration is another important feature of the multiprocessor architecture since a system like Cognitive Radio needs to adapt dynamically to a changing environment in real time. The combination of high-level design and run-time tools with reconfigurable hardware architectures will enable designers to develop highly flexible, efficient and adaptive systems and applications for future systems such as Cognitive Radio.

A multi-core architecture has to be supported by a predictable intercore communication network e.g. NoC. A NoC that routes data items has a higher bandwidth than an on-chip bus, as it supports multiple concurrent communications. The well-controlled electrical parameters of an on-chip interconnection network enable the use of high-performance circuits that result in significantly lower power dissipation, higher propagation velocity and higher bandwidth than is possible with conventional circuits [44].

Among the heterogeneous processing elements, we consider reconfigurable architectures, particularly coarse-grained reconfigurable architectures (e.g. Montium), as the most important element on the MPSoC for SDR and Cognitive Radio. As we have discussed in the previous section, reconfigurable hardware offers good tradeoffs in performance, flexibility and power consumption. In [67], the author proposed a similar view for the future trend of MPSoC based SDR. Reconfigurable architectures and DSP-centered and accelerator-assisted architectures are thought to be two good candidates for future SDR.

2.3.2 Case Study: Annabelle SoC

Based on the template architecture in figure 2.1, a SoC called Annabelle (see figure 2.2) has been develop in our group and was processed with ATMELs

2.3 Heterogenous Reconfigurable Multiprocessor System-on-Chip

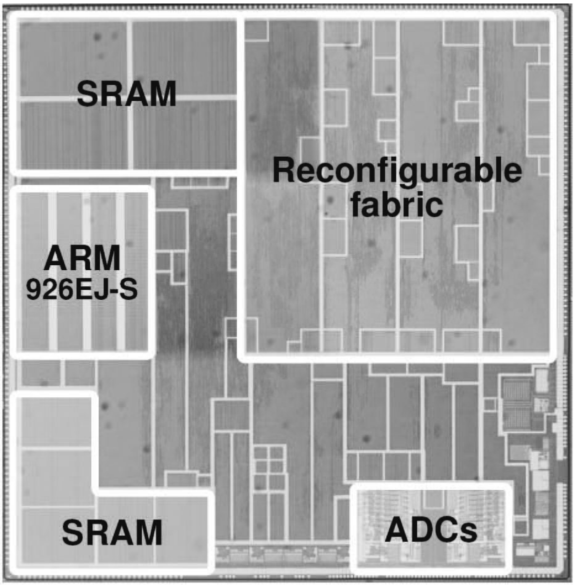

Figure 2.2: A picture of the Annabelle chip

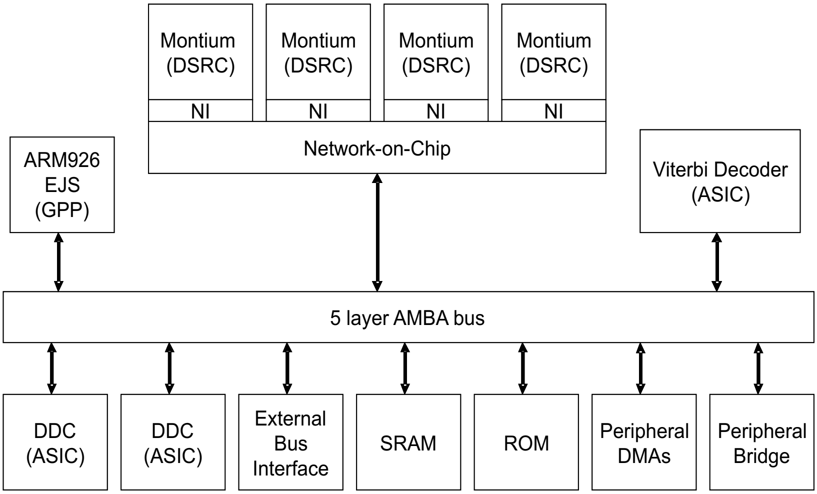

Figure 2.3: Block diagram of the Annabelle chip

proprietary process (130 nm process). In the Annabelle SoC (see the block diagram in figure 2.3) a conventional ARM926 architecture is complemented by ASIC blocks (for example Viterbi and DDC) and four domain specific coarse-grain reconfigurable Montium cores. A predictable circuit-switched NoC [107] interconnects the four Montium cores via the network interfaces (NI). The ARM926 processor serves as the control processor by sending configuration messages. All components are connected via the Advanced Microcontroller Bus Architecture (AMBA) system.

In the end of 2007, we received the first samples of the chip. The ASIC synthesis results (see details in section 2.4) of the Annabelle shows that this Montium based MPSoC is very energy effcient.

2.4 Coarse-grained Reconfigurable Architectures

As discussed in section 2.2, coarse-grained reconfigurable hardware is considered as a key element in future SDR because of its flexibility, performance and energy efficiency in general. As an example of coarse-grained reconfigurable hardware, the Montium is mentioned in section 2.3. The Montium tile processor is developed in our group at University of Twente [39]. Recently, it has been commercialized by a spin-off company named Recore Systems[2]. As a key element of our propose an MPSoC platform for Cognitive Radio, the Montium is the targeted coarse-grained reconfigurable architecture for the discussion throughout the thesis. In this section, we will give an introduction on the Montium followed by a brief overview on related work. More details can be found in [39].

2.4.1 The Montium Architecture

The Montium tile processor is targeting the 16 bit Digital Signal Processing (DSP) algorithm domain. A single Montium tile processor is depicted in figure 2.4. At first glance the Montium architecture bears a resemblance to a Very Long Instruction Word (VLIW) processor. However, the control structure of the Montium is optimized to minimize the control overhead which is imperative for energy efficiency. The Montium consists of two major parts: the tile Processor (TP) and the Communication and Configuration Unit (CCU).

[2] Recore Systems website: `www.recoresystems.com`

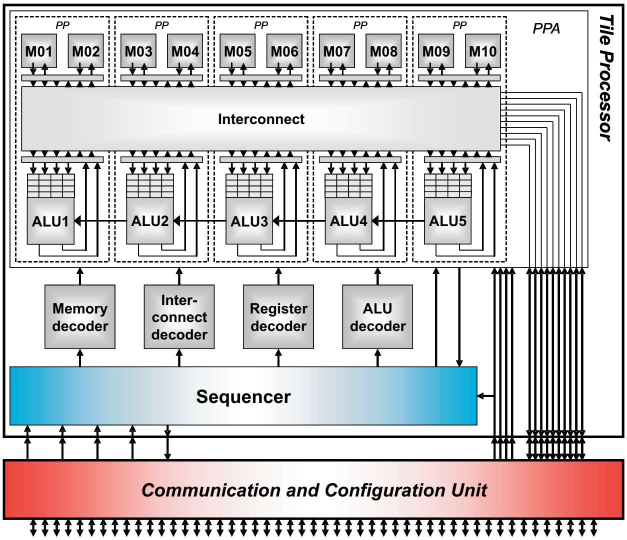

Figure 2.4: The Montium tile processor

Tile Processor

The computing part, the Tile Processor (TP) can be configured (reconfigured) to perform a particular algorithm. The five identical Arithmetic Logic Units (ALU1...ALU5) in a tile can exploit spatial concurrency to enhance performance. The data path of the ALUs has a width of 16-bits and the ALUs support both signed integer and signed fixed-point arithmetic. The parallelism demands a very high memory bandwidth, which is obtained by having 10 local memories (M01...M10) in parallel. The local memories imply a good locality of reference. An ALU and its two local memories form a processing part (PP). Five such PPs makes a processing part array (PPA). A relatively simple sequencer controls the entire tile processor. The sequencer selects configurable tile instructions that are stored in the decoders (see figure 2.4).

Each local memory is 16-bit wide and has a depth of 1024 positions, which adds up to a storage capacity of 16Kbit. A reconfigurable Address Generation Unit (AGU) accompanies each memory. The AGU can generate the most frequently used address patterns, but when needed also an ALU can generate address patterns. It is also possible to use the memory as a lookup table for complicated functions that cannot be calculated using an ALU, such as sine or division (with one constant). A memory can be used for both integer and fixed-point lookups.

Each one of four 16-bit inputs to an ALU has a private input register file that can store up to four operands. The input register file cannot be bypassed and an operand is always read from an input register. Input registers can be written by various sources via a flexible interconnect. Two 16-bit outputs from an ALU are connected to the interconnect. Neighboring ALUs can also communicate directly: the West-output of an ALU connects to the East-input of the ALU neighboring on the left. The ALU is entirely combinational and consequently there are no pipeline registers within the ALU. It can implement the general arithmetic and logic operations as well as the Multiply-Accumulate (MAC) typically in DSP.

Communication and Configuration Unit

The CCU [9] is an interface developed to serve the Montium for off-tile communication. It provides the functionality to configure the Montium, to manage the memories by means of direct memory access (DMA) and to control the computation of the algorithm configured. Both streaming-mode and block-mode communication are supported by the CCU. In block-mode, the DMA interface is used to manage data transfers. The CCU acts as a master for the tile processor, while the data transfers are directed by the control processor. In streaming-mode the algorithm designer implicitly describes how the data transfers are performed and, therefore, the Montium controls the CCU to take care of the data flow from and to the NoC.

Tooling Support

Good tooling support is essential for developers to quickly and efficiently implement applications on reconfigurable architectures. The tool development for the Montium has been undertaken in parallel with the development of the architecture. In [29], a framework for efficient compilation of applications written in high-level languages (C/C++) to the Montium was proposed. To efficiently implement applications on the Montium, an assembly-like language called Montium Configuration Description Language (CDL) is used. The Montium tool set also includes a stand-alone Montium simulator, a Montium assembler and a Montium configuration editor. The tool development for the Montium is still a ongoing effort at the Recore Systems.

Montium Applications

Since the Montium is targeted for low-power DSP processing, mapping of computational intensive DSP algorithms in wireless communication onto the Montium has been one of our major focuses. In previous work [70], key algorithms in several popular wireless standards such as HiperLAN/2 and WCDMA have been implemented on the Montium. The Montium is flexible enough to adapt to different algorithms. It can perform not only algorithms with regular patterns such as radix-2 FFT but also algorithms with irregular patterns such as non-power-of-two FFTs [73]. Moreover, it can perform bit oriented coding schemes such as Viterbi and Turbo codes [69]. Supporting emerging baseband processing algorithms for Cognitive Radio on the Montium is one of the topics of this thesis.

Synthesis Results on the Annabelle MPSoC

An ASIC synthesis of the Annabelle was performed using ATMELs proprietary 130 nm process technology. For the local data memories and sequencer instruction memory of the Montium TP, embedded SRAMs are used. The embedded SRAM is an optimized component from a cell library. The worst case military conditions are assumed. In particular, the supply voltage is 1.1 V and the temperature is 125 ^oC.

Table 2.1: Static power consumption of one Montium on Annabelle

Module	Static power [mW]
Datapath	0.09
Memories	0.06
Sequencer	0.01
Decoders	0.03
CCU	0.01
Total	0.20

To determine the average power consumption of the Annabelle, a number of power estimations are performed on the placed and routed netlist using the Synopsis [90] power compiler. Table 2.1 gives the static power consumption of various Montium blocks. Table 2.2 provides the dynamic power consumption in mW/MHz of various Montium blocks for an FFT-512 (denoting a 512 point FFT). Finally, the energy consumption of several DSP algorithms on the Montium and on the ARM926 on Annabelle are

Table 2.2: Dynamic power consumption of one Montium on Annabelle

Mondule	FFT-512 [mW/MHz]
Datapath	0.24
Memories	0.27
Sequencer	0.07
Decoders	0.0
CCU	0.02
Total	0.60

Table 2.3: Energy comparison Montium/ARM926

	Montium	ARM926
Algorithm	[μJ]	[μJ]
FIR-5	0.243	-
FFT-512	1.563	30
FFT-1920	5.054	168

compared in Table 2.3. The area of one Montium core is 3.5 mm^2 of which 0.2 mm^2 is used for the CCU and 3.3 mm^2 for the Montium TP. Within the Annabelle ASIC realization, the clock speed of the Montium can be as high as 100 MHz and 25 MHz in the worst case.

From these results [82], we can see that the Montium is very energy efficient for its algorithm domain. Particularly, the overhead of the sequencer and decoder is low: e.g. for an FFT-512 less than 12% of the total dynamic power consumption. Therefore, with its energy efficiency and reconfigurability, the Montium can be a good candidate to support the baseband processing of Cognitive Radio.

2.4.2 Other Coarse-grained Reconfigurable Architectures

Many research efforts have been undertaken to develop coarse-grained reconfigurable architectures in both academia and industry. Here a brief overview of related work on coarse-grained reconfigurable architectures is given. More comprehensive overviews including related references can be found in [93] and [35].

The Pleiades architecture

The Pleiades architecture was proposed by the University of California at Berkeley targeting DSP algorithms [1]. In the Pleiades architecture (see figure 2.5), a control processor, usually a GPP, is surrounded by a heterogeneous array of satellite processors (SPs). All processors communicate over a reconfigurable communication network. The control processor controls and configures the satellite processors and the communication network at run-time. The satellite processors can be optimized for specific algorithm domains such as computationally intensive DSP algorithms. An example of such satellite processor in [1] is a processor called Maia, which is six times more energy-efficient than the DSP solution for executing a speech coding algorithm. This idea is quite similar to the motivation of the Annabelle heterogenous MPSoC architecture.

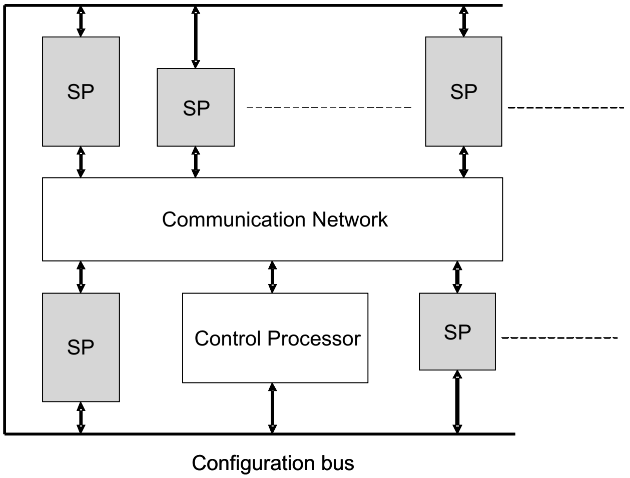

Figure 2.5: Block diagram of the Pleiades architecture

The MorphoSys reconfigurable processor

At the University of California, Irvine, the MorphoSys reconfigurable processor [78] was developed for data-parallel and computation intensive applications. The MorphoSys architecture (see figure 2.6) consists of an array of reconfigurable processors with configuration memory to exploit the parallelism in a given application. One reconfigurable processor has an ALU with multiplier, a register file and is configured through a 32-bit context word. The reconfigurable processor array is connected to the system bus with a high bandwidth data interface. A Reduced Instruction Set Computer (RISC) processor serves as a control processor to handle serial operations, initiate data transfers and controls operation of the reconfigurable processor array. The external data memory can be accessed by a DMA controller. The MorphoSys has larger silicon area than the Anabelle chip. The reconfigurable processor array is configured by context words, which specify an instruction opcode for the array.

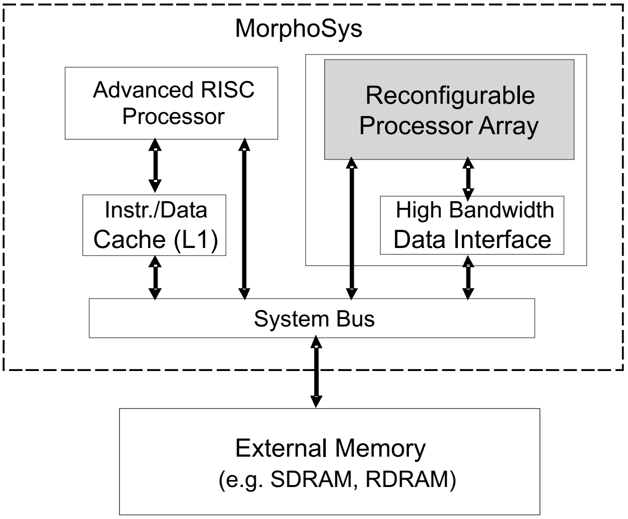
MorphoSys
Advanced RISC Processor
Instr./Data Cache (L1)
Reconfigurable Processor Array
High Bandwidth Data Interface
System Bus
External Memory (e.g. SDRAM, RDRAM)

Figure 2.6: Block diagram of the Morphosys architecture

The eXtreme Processing Platform

The eXtreme Processing Platform (XPP)(see figure 2.7) [4] is a run-time reconfigurable data processing platform. The XPP is based on an array of coarse-grained, adaptive computing elements called Processing Array Elements (PAEs) connected through a packet-oriented communication network.

An XPP device may contain one or several Processing Array Clusters (PACs). Each PAC is controlled by a Configuration Manager (CM) which writes configuration data into the configurable object of the PAC. In a Multi-PAC XPP platform, there is a Supervising Configuration Manager (SCM) to handle concurrent configuration data on top of other CMs. Each PAC contains arrays of Processing Array Elements (PAEs). A PAE can be either an ALU PAE or a memory PAE. The ALU PAE contains a multiplier and is used for DSP operations. The memory PAE contains a Random Access Memory (RAM) to store data. PAEs can be configured rapidly in parallel while neighboring PAEs are processing data. Entire applications can be configured and run independently on different parts of the array. Compared with the Annabelle platform, the PACT-XPP has a larger silicon footprint. The control of the PACT-XPP is data-flow like.

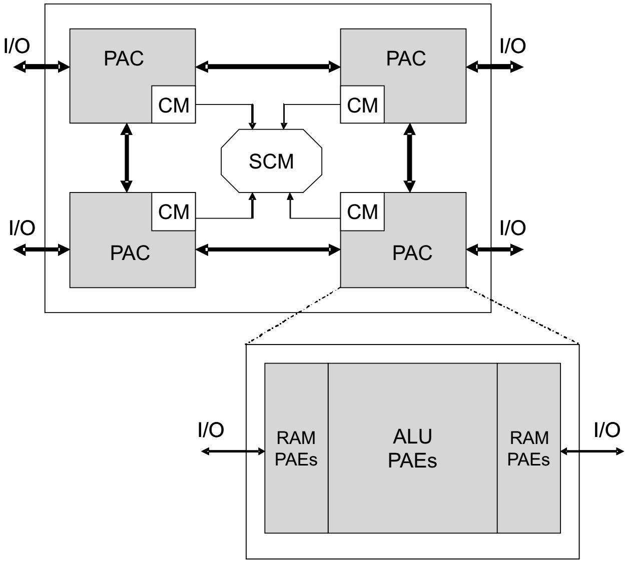

Figure 2.7: Block diagram of XPP processing platform

2.5 Design Methodology

To map an application on a hardware platform, the design methodology is essential. A good design methodology should have the following attributes:

- The mapping of the application on the platform should be nearly optimal from a resource consumption point of view.

- The design cycle should be short.

- The solution should adhere to the QoS requirements (i.e. throughput and latency of the application).

As we have discussed, a heterogenous MPSoC platform is a good candidate platform to support Cognitive Radio. Therefore, a design methodology for mapping Cognitive Radio algorithms onto an MPSoC platform is required.

2.5.1 Design Challenges for MPSoC platforms

MPSoCs offer many advantages such as flexibility and energy efficiency. However, it is a challenging task to map applications onto MPSoCs.

First, the applications to be mapped become increasingly complex and may change their behavior dynamically. If Cognitive Radio is considered, it has more functionalities than traditional radios and it changes its behavior dynamically. The complexity and dynamism make Cognitive Radio difficult to be mapped onto the targeted platform.

Second, in order to map tasks to different components on an MPSoC, designers have to deal with the low level interfaces for the inter-component communication and synchronization which become a bottleneck from a performance and an energy point of view.

Further, opportunities for reuse of hardware and software modules are limited and a design methodology for exploring their trade-offs is missing.

Therefore, there is a gap between the application models used for specification and the optimized implementation of the application on an MPSoC. A task transaction level (TTL) interface approach [106] was proposed to help to close the gap by raising the abstraction level of application programs. We propose to use the TTL approach both for developing the Cognitive Radio application at the system level and as a platform interface for implementing the application onto the MPSoC architecture.

2.5.2 The TTL Interface

DSP applications are often modelled as streaming task graphs which consist of logical entities like tasks, ports and channels as shown in figure 2.8 [106]. *Tasks* are entities that perform data processing and they can communicate with other tasks by sending data via *ports* to *channels* which interconnect tasks. A channel is used to transfer *values*. A value is held in a *variable*, a logic storage location. Private variables can only be accessed by one task. A variable which holds a value to be communicated from one task to another is called *token*. A token is full if it contains a value, otherwise it is empty. Empty and full tokens are referred to as *data* and *room*, respectively. The TTL communication interface is based on these logical models. Intertask

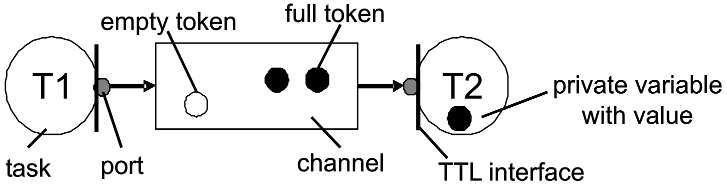

Figure 2.8: The TTL logic model

communications are invoked by calling TTL interface functions, like *read* or *write* on their ports. The TTL interface can support different communication styles by providing a set of different interface types [106]. In this thesis, only one type of interface, Combined Blocking (CB), is considered. The CB interface combines synchronization and data transfer in a single function with blocking semantics.

The TTL interface separates computation from communication so that computation can be implemented easier onto on particular tile processors. It also provides high level profile information in terms of computation workload and communication workload. In such a way, we can explore and validate design alternatives at a high level of abstraction. Subsequently, the high level TTL specification will be mapped onto a multiprocessor architecture.

In [106], only static task graphs were considered. However, embedded systems are becoming more and more complex, supporting multiple use cases. Therefore, TTL has been extended to model dynamic applications [43]. Dynamic applications result in changes in the task graph. Three types of task graph changes are considered: 1) Topology: removing or adding tasks and channels, 2) Binding: tasks may be processed on different processors, 3) Parameters: certain parameters for tasks (e.g. radio transmission modes). The reconfiguration is realized by introducing an extra entity, a Configuration Manager (CM), on top of the TTL model. The CM is responsible for initiating the task graph and for configuring the tasks at runtime. This feature makes TTL a suitable design method to model dynamic applications such as Cognitive Radio and as an interface for the MPSoC platform implementation. Therefore, the TTL interface is chosen as the system-level design methodology for mapping Cognitive Radio to the proposed MPSoC platform. The detailed approach of TTL for mapping Cognitive Radio onto the proposed platform and its benefits will be elaborated by specific design examples of Cognitive Radio in the subsequent chapters.

2.6 Chapter Summary

This chapter mainly focuses on the hardware architecture for a SDR platform that will eventually evolve to Cognitive Radio. Several common embedded computer architectures have been introduced and their advantages and disadvantages for SDR have been discussed:

- General Purpose Processors (GPPs) are very flexible and easily programable to execute any algorithm. However, they are not efficient in terms of performance and energy efficiency for computationally intensive tasks such as DSP algorithms in wireless baseband processing.

- Digital Signal Processors (DSPs) can be seen as a type of GPPs with enhanced features for digital signal processing. However, fetching and decoding instructions still incur a large overhead.

- Application Specific Integrate Circuits (ASICs) are the most efficient solutions regarding performance and energy efficiency. However, due to their inflexibility, they are unable to fulfil dynamic reconfiguration requirements of Cognitive Radio.

- Reconfigurable hardware combines flexibility, performance and energy efficiency. Reconfigurable hardware architectures and especially coarse-grained reconfigurable architectures are considered to be the prime candidate for future SDR.

We expect that heterogenous reconfigurable Multiprocessor System-on-Chip (MPSoC) platforms are the future for SDR. A tiled MPSoC platform is proposed to support the adaptable physical layer of Cognitive Radio. As we have direct access to the hardware architecture and tool support of the Montium, a coarse-grained reconfigurable processor developed in our group, it is the key element in our proposed MPSoC platform. The feature of the Montium includes:

- Parallel computing units;

- Minimum control overhead;

- Memory locality;

- High internal bandwidth

- Combination of flexibility, performance and energy efficiency.

Although most experiments are done with the Montium, we believe the results are more generally applicable for coarse grained reconfigurable architectures.

Finally, we introduce a design methodology for mapping the adaptable physical layer of Cognitive Radio onto the proposed MPSoC platform. The design methodology is based on the task transaction level (TTL) interface approach [106]. The choice of TTL for Cognitive Radio design on the MPSoC is based on the following major features that it offers:

- it helps to close the gap between the application models used for specification and the optimized implementation of the application on an MPSoC by raising the abstraction level
- it provides high level profile information in terms of computation workload and communication workload.
- it can model dynamic applications such as Cognitive Radio

The TTL approach will be used throughout the thesis as the system-level design methodology and its advantages will be further elaborated by the specific design examples of Cognitive Radio.

The preliminary conclusion we can draw from this chapter is that an MPSoC based SDR platform will be the prime candidate platform for Cognitive Radio. The key elements in such a platform are reconfigurable hardware building blocks, especially coarse-grained reconfigurable elements. The TTL approach is a suitable design methodology for mapping Cognitive Radio on such a platform.

Chapter 3

OFDM Based Cognitive Radio

The choice of the physical layer transmission scheme is a primary design decision for Cognitive Radio. The physical layer of Cognitive Radio has to offer agility to access the licensed band on a noninterference basis and make best use of available spectrum. Like other modern wireless multimedia systems, the physical layer of Cognitive Radio should be able to provide high data rates. The family of multicarrier based transmission techniques has been widely recognized as one of the best candidates for Cognitive Radio to fulfil these requirements. The most common multicarrier transmission scheme, OFDM, is discussed in this chapter[1] *as the physical layer transmission scheme for Cognitive Radio.*

[1] Parts of this chapter have been published in publications [QZ4] [QZ5] [QZ9].

3.1 Introduction

The choice of a physical layer transmission scheme is a primary design decision for implementing Cognitive Radio. In dynamic spectrum Cognitive Radio, there are several requirements concerning the physical layer transmission scheme. Firstly, the technique should be sufficiently agile to enable Cognitive Radio to access the licensed band on a non-interference basis. Secondly, the technique should enable Cognitive Radio to optimally use the available spectrum resources even in case the spectrum is segmented. Moreover, like other modern wireless multimedia systems, the technique should be able to provide high data rates. To meet all the requirements, the family of multicarrier based transmission techniques has been widely recognized as the best candidate. In the context of our AAF project, we also consider the use of multicarrier techniques for the physical layer transmission scheme for Cognitive Radio based emergency networks.

The basic idea of multicarrier transmission is to divide the data streams into several parallel substreams, each with a lower data rate, and modulate these substreams with different subcarriers [60]. As a result, multicarrier modulation is more robust to frequency selective fading and narrow band interference compared to single carrier modulation. We show an example of the channel effects on multicarrier modulation and single carrier modulation in figure 3.1. Suppose the total bandwidth is B, the bandwidth of each subband in the multicarrier system with N subcarriers is $\frac{B}{N}$. In the single carrier system, a single deep fade (as indicated in figure 3.1) may cause a failure of the entire transmission. However, in a multicarrier system, only a certain percentage of the subcarriers will be affected. Therefore, multicarrier modulation is considered to be a good candidate for high data rate broadband transmissions.

In the context of Cognitive Radio, subcarriers of multicarrier systems can be switched off to avoid interference to licensed users. This idea is also known as spectrum pooling [103]. One of the most widely used multicarrier techniques, Orthogonal Frequency Division Multiplexing (OFDM) has been considered as the physical layer transmission scheme for spectrum pooling systems. However, due to its spectrum leakage problem, traditional OFDM can not be used for Cognitive Radio without modifications.

Section 3.2 introduces some fundamentals of OFDM including some example OFDM systems. The design of the baseline OFDM system in the context of AAF is also considered. Section 3.3 discusses the OFDM system for Cognitive Radio. Two general principles apply to OFDM based Cognitive Radio: 1) avoid interference to licensed users and 2) make optimal use

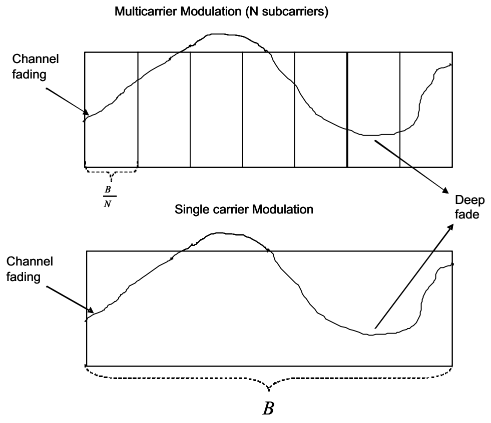
Multicarrier Modulation (N subcarriers)
Channel fading
B/N
Single carrier Modulation
Deep fade
Channel fading
B

Figure 3.1: The effects of channel fading on multicarrier modulation and single carrier modulation

of the available spectrum. Based on these principles, we propose an adaptive OFDM system for Cognitive Radio. As the first step in the process of mapping the proposed OFDM based Cognitive Radio onto the reconfigurable platform, the adaptive OFDM system is modelled at system level with the TTL approach introduced in section 2.5. The profile information generated by the TTL model helps us at an early design stage to identify where the system complexity is. The TTL approach helps to partition the computation of the adaptive OFDM system into tasks which can be treated independently, thus facilitates the mapping onto the proposed MPSoC. Finally, the chapter is summarized and some conclusions are drawn in section 3.5.

3.2 OFDM Based Transmission

OFDM is the most commonly used multicarrier transmission scheme today. The idea of OFDM was first proposed more than 40 years ago by Chang [13]. Despite its conceptual beauty, the actual implementation was extremely difficult in the early day's analog communication systems. With the maturity of digital electronics and digital signal processing techniques, the Discrete Fourier Transform (DFT) was proposed as the digital modulation technique for OFDM systems by Weistein et al. [101]. By applying the Fast Fourier Transform (FFT) technique, the computational cost of DFT can be further reduced. Recent advances in very-large-scale integration (VLSI) technology makes the FFT based OFDM solution commercially affordable. Therefore, OFDM has been adopted in many standards such as Digital Audio Broadcasting (DAB) [22], Digital Video Broadcasting (DVB) [24], Digital Radio Mondial (DRM) [23] and Digital Subscriber Line (DSL) [3] to provide high data rate communications.

3.2.1 Fundamentals of OFDM

The basic principle of OFDM is to divide a high-rate data stream into a number of low-rate substreams which are modulated with orthogonal subcarriers. The orthogonality makes OFDM differ from the traditional Frequency Division Multiplexing (FDM) in the sense that OFDM allows overlapped subcarriers. These overlapped subcarriers result in more efficient spectrum usage. Therefore, OFDM can be considered as a spectrally efficient version of FDM. In a classical method [13], these orthogonal subcarriers are generated by an array of passband filters. Weinstein et al. discovered that the orthogonal frequencies can be generated by the DFT which consists of a set of orthogonal sinusoids [101]. Furthermore, the DFT can be computed efficiently with an FFT. As a result, OFDM modulation is based on the FFT.

Basic Scheme

A generic block diagram of a basic OFDM system, based on the IFFT/FFT algorithm is shown in figure 3.2. First, a high rate bitstream is split into N low rate streams. These streams are then individually modulated by either Quadrature Amplitude Modulation (QAM) or Phase Shift Keying (PSK) to form N parallel streams with complex symbols. We use $X_{m,k}, k = 0, 1, ...N - 1$ to denote the complex symbols of the k-th stream at the m-th time instant. The symbol streams are modulated with N orthogonal subcarriers and summed up to yield OFDM symbols. Following the parallel-to-serial conversion, the OFDM signal is upsampled and converted to an analog signal. Then the baseband signal is upconverted to the desired centering frequency for transmission. The heart of the scheme, the modulation with N orthogonal subcarriers, can be realized by an Inverse Discrete Fourier Transform (IDFT) which can be implemented as an Inverse Fast Fourier Transform (IFFT). If we consider the discrete samples, the OFDM symbol can be written as:

$$s_m(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_{m,k} e^{\frac{j2\pi kn}{N}} \quad n = 0, 1, ..., N - 1 \tag{3.1}$$

A guard interval that is larger than the expected channel delay spread is

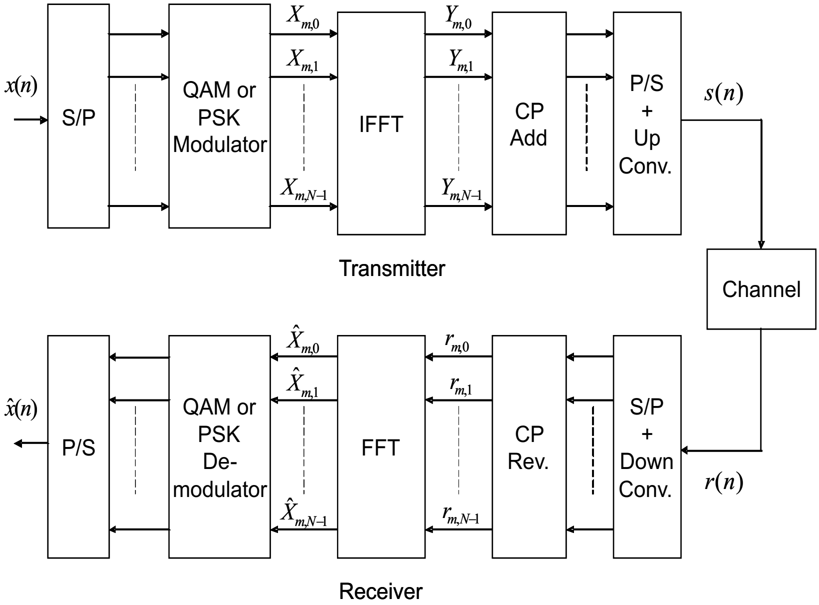

Figure 3.2: General block diagram of a basic OFDM transceiver

often added to each OFDM symbol to guarantee that multipath components from one symbol will not interfere with the next symbol. The guard interval is usually a cyclic extension of the OFDM signal, also known as cyclic prefix(CP Add). This means that the guard interval consists of the copy of the last N_g samples of the time domain OFDM signal. Figure 3.3 shows the cyclic extension of one OFDM symbol. The main reason for a cyclic extension is to make the channel effects equivalent to a cyclic convolution. Due to the properties of the cyclic convolution, the inter-symbol interference (ISI) is eliminated to a large extent and inter-carrier interference (ICI) can be manageable with simple frequency domain equalization. Furthermore, the cyclic extension is also useful for synchronization purposes since the cyclic extension introduces periodicity. Obviously, the drawback of introducing a cyclic prefix is the loss of bandwidth efficiency.

The receiver performs the reverse operation of the transmitter. The received signal is downconverted to baseband and digitized. Then the signal is converted into parallel streams by a serial-to-parallel (S/P) converter and the cyclic prefix is discarded(CP Rev.) from the received signal. Then, the

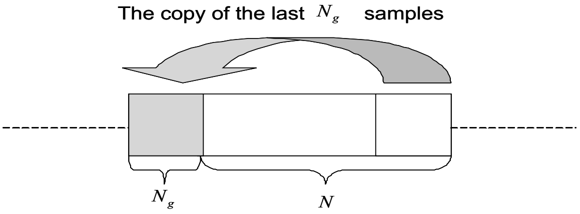

Figure 3.3: The cyclic extension of the OFDM symbol

DFT is used to transform the time domain data into frequency domain:

$$\hat{X}_{m,k} = \sum_{n=0}^{N-1} r_{m,n} e^{\frac{-j2\pi kn}{N}} \quad k = 0, 1, ..., N-1 \tag{3.2}$$

An efficient implementation of the DFT is the FFT. After the FFT, parallel streams are converted to a single serial stream for demodulation into a bit stream.

System Design Considerations

To design an OFDM system, several basic parameters should be taken into consideration: the bandwidth of the OFDM signal B, the number of subcarriers N and the length of cyclic prefix N_g. The choice of the parameters largely depends on the channel conditions under which the OFDM system is operating. Two parameters are often used to characterize the channel: the delay spread τ and the maximum doppler spread f_d. The delay spread characterizes the time difference between the Line-of-Sight (LOS) signal and its multipath components from the channel. The multipath components cause inter-symbol interference (ISI) when the symbol prefix duration is shorter than the delay spread. The maximum doppler spread is inversely related to the channel coherence time during which the channel is considered to suffer a time invariant fading.

First of all, the size of the cyclic prefix or guard time $T_g = N_g T$ ($T = \frac{1}{B}$ denotes the sampling time) has to be larger than the delay spread τ to eliminate ISI. To limit the bandwidth loss due to the guard time, the useful time of the OFDM symbol $T_u = NT$ has to be much larger than the delay spread, that is $NT \gg \tau$. On the other hand, the useful symbol time T_u should be much shorter than the channel coherence time. If we take the inverse of the doppler spread $\frac{1}{f_d}$ as the measure of the coherence time, T_u should satisfy $T_u = NT \ll \frac{1}{f_d}$. These two constraints give the relation:

$\tau \ll NT \ll \frac{1}{f_d}$. If we relate the number of subcarriers N to the bandwidth B, the following relation can be obtained:

$$\tau B \ll N \ll \frac{B}{f_d} \tag{3.3}$$

This relation serves as a guiding principle to select the number of the subcarriers N. If the bandwidth and the number of subcarriers are determined, the subcarrier spacing Δf is also determined as $\Delta f = \frac{B}{N}$. In such a way, a set of basic parameters for an OFDM system can be derived. In practical systems, there may be several parameter sets defined for the systems to adapt to various channel conditions. For example, the ratio of the guard samples to the useful samples $N_g : N$ can be different to adapt to channels with various delay spreads. When considering an OFDM system for Cognitive Radio, we foresee that an adaptive parameter set will be one of the important features. Furthermore, for implementation purposes, N is often rounded to a power-of-two integer to take advantage of a radix-2 FFT.

Open Issues

In general OFDM systems, there are several open issues which make up the main research topics for an OFDM physical layer. These challenges are related to channel distortions and the characteristics of the OFDM signal.

- **Synchronization**

  Communication systems are subject to timing and frequency offsets due to oscillator impairments and sample clock differences. In single carrier systems, timing and frequency offsets will only cause performance degradation. However, an OFDM system is rather sensitive to frequency offsets and phase noise. The reason is that the subcarriers are perfectly orthogonal only if transmitter and receiver use exactly the same frequencies. Any frequency offset and phase noise will destroy the orthogonality between subcarriers causing intercarrier interference (ICI). The sensitivity to frequency offset and phase noise is often mentioned as a disadvantage of OFDM systems compared to single carrier systems [64]. Therefore, to mitigate the ICI caused by frequency offset and phase noise, the OFDM receiver has to estimate and compensate frequency offset and phase noise. In general, estimation methods are either based on pilot information or take advantage of statistical redundancy introduced by the cyclic prefix [60].

- **Equalization**

  By introducing a proper cyclic prefix, the inter-symbol interference (ISI) can be largely eliminated. The channel effect is equivalent to a circular convolution. By taking the FFT on the circular convolution, a wideband channel is translated into a number of flat fading subchannels. The effect of subchannel fading can be seen as the multiplication of the symbol on each subcarrier with a complex-valued attenuation. The channel equalizer has to track, estimate and compensate the fading on each subchannel. Usually, channel estimation in OFDM is performed with the aid of pilot symbols. The simplest estimator just divides the received pilot symbol on each subcarrier by the original pilot symbols (known at the receiver) to obtain an estimate of the channel attenuation. This method is also known as zero forcing [68]. However, zero forcing will amplify noise particularly in a deep attenuation case. The Minimum Mean Squared Error based channel estimator is optimal in performance [68], however it is of little practical value due to its computational complexity. Therefore, the main challenge of channel equalization in OFDM systems is to find high performance but low-complexity estimator algorithms.

- **Channel Coding**

  Channel coding is an essential part of an OFDM system for mobile communication. OFDM in a fading environment is almost always used in conjunction with channel coding to improve its performance and is therefore often referred to as Coded OFDM (COFDM) [120]. Since burst errors degrade the performance of the channel decoder, COFDM is often accompanied by frequency (subcarrier) interleaving which makes the bit errors in the faded part less concentrated. Therefore, when designing a practical OFDM system, the interleaving and channel coding scheme always have to be considered as well.

- **Peak-to-Average Power Ratio**

  Since an OFDM signal consists of a number of independently modulated subcarriers, high power peaks can occur when added up coherently. This property is often measured via the signal's peak-to-average power ratio. A large peak-to-average power ratio imposes high demands on analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) in terms of range and precision. It also reduces the efficiency of the RF power amplifier. Reducing the peak-to-average power ratio is another challenge for OFDM. Generally, there are three concepts for reducing the peak-to-average power ratio [60]. The first concept is to nonlinearly distort the signal at or around the peaks. The second concept reduces the peak-to-average power ratio by designing a coding scheme to exclude the OFDM symbols with large peak-to-average power ratios. The third one is based on a scrambling sequence.

To overcome all these challenges, a practical OFDM system often requires additional signal processing blocks such as frequency offset correction, channel equalization and peak-to-average power reduction on top of the basic system shown in figure 3.2. These additional processing blocks add more complexity to the OFDM system. Moreover, some bandwidth has to be reserved for carrying pilot information. This also reduces the efficiency of the OFDM system.

When considering the OFDM system for Cognitive Radio, the aforementioned issues also exist and need to be solved. Apart from these, additional issues have to be addressed for the OFDM system in the context of Cognitive Radio. One of the biggest issues is the large power leakage of the OFDM system. We can foresee that the OFDM system for Cognitive Radio will be more adaptive and hence more complex. The discussion of the OFDM system for Cognitive Radio will be carried on in section 3.3. In chapter 5, the problem of power leakage will be discussed in more detail.

3.2.2 Overview of OFDM Systems

Recently, OFDM has been adopted by many standards as a physical layer scheme. OFDM based electronic products become quite common. In future communication systems, OFDM will play a major role. Here, a short overview is given to introduce several types of OFDM based communication systems.

High Performance Radio LAN type 2 (HiperLAN/2) [21] is a wireless LAN (WLAN) access technology proposed by the European Telecommunications Standards Institute (ETSI). HiperLAN/2 operates in the 5 GHz frequency band. The air interface is based on Time Division Multiple Access (TDMA) and Time Division Duplex (TDD). OFDM is used as the physical layer transmission scheme. The major parameters are summarized in Table 3.1. The HiperLAN/2 OFDM signal occupies a 20 MHz bandwidth B. It consists of 64 subcarriers which indicates the FFT size is 64. 48 out of 64 subcarriers are used for carrying data symbols (defined as K) while others are used for carrying pilot information or are just unmodulated. The useful time T_u in one OFDM symbol is 3.2 μs and the guard time T_g is 0.8 μs, which together make the whole OFDM symbol duration T_{sym} equal to 4 μs. The ratio $T_u : T_g = 4 : 1$ implies the number of guard samples is 16. The modulation types applied on each subcarrier can be Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), 16 Quadrature Amplitude Modulation (16QAM) and 64QAM. Different channel coding rates can be applied to offer levels of robustness. The achievable data rate ranges from 6 Mbps to 54 Mbps.

Table 3.1: The major parameters of the OFDM system for HiperLAN/2

B $[MHz]$	N	K	T_{sym} $[\mu s]$	T_g $[\mu s]$	T_u $[\mu s]$	Δf $[kHz]$
20	64	48	4	0.8	3.2	312.5

Table 3.2: The major parameters of the OFDM system for DAB with different modes

Mode	B $[MHz]$	N	K	T_{sym} $[\mu s]$	T_g $[\mu s]$	T_u $[\mu s]$	Δf $[kHz]$
I	1.54	2048	1536	1246	246	1000	1
II	1.54	512	384	312	62	250	4
III	1.54	256	192	156	31	125	8
IV	1.54	1024	768	623	123	500	2

Table 3.3: The major parameters of the OFDM system for DRM with different modes

Mode	B $[MHz]$	N	K	T_{sym} $[\mu s]$	T_g $[\mu s]$	T_u $[\mu s]$	Δf $[kHz]$
A	0.01	288	225	26667	2667	24000	0.0417
B	0.01	256	205	26667	5333	21333	0.0469
C	0.01	176	137	20000	5333	14667	0.0682
D	0.01	112	87	16667	7333	9333	0.1071

The OFDM based physical layer is also adopted by many broadcasting standards. The Digital Audio Broadcasting (DAB) standard [22] is the replacement of the analog FM radio services and has been adopted by many countries around the world. The DAB radio system can provide high-quality audio services with a maximum data rate of 1.8 Mbps. The DAB standard has defined different parameter sets for different transmission modes to adapt to different channel conditions (see Table 3.2). The DAB system occupies a bandwidth of 1.54 MHz. FFT sizes of 2048, 512, 256 and 1024 are used for four different transmission modes. Only 1563, 384, 192 and 768 carriers are modulated. One of the important features in the DAB system is that the subcarrier modulation applies Differential QPSK (DQPK). Such a differential modulation scheme avoids channel equalization, thus results in a simple receiver structure and the avoidance of pilot symbols.

The Digital Radio Mondiale (DRM) [23] has been proposed to replace the analog AM radio service. The idea of the DRM is to coexist with the existing AM radio service so that it has to fit within the existing AM broadcast band plan. Even with a limited bandwidth of only 10 kHz, it still delivers high quality MPEG-4 audio. The OFDM system, based on different parameter sets, is used for the physical layer transmission (see Table 3.3). It is interesting to look at the relation between the choice of the mode and the transmission mode. Mode A is intended for a Gaussian channel with very little multipath propagation and doppler effects. Therefore, it has the smallest subcarrier spacing Δf and the shortest guard time T_g. Mode B is intended for the typical multipath fading channel. Compared to Mode A, Mode B has a longer guard time to counter delay spreads and smaller carrier spacing to mitigate the doppler effect. Mode C has better robustness to doppler effects than Mode B by increasing the subcarrier spacing. A larger subcarrier spacing results in a shorter symbol duration which is less subject to time variations caused by doppler effects. Mode D targets the worst case scenario: a channel with large delay spread and doppler spread. As a result, it has the longest guard time and the largest subcarrier spacing. However, all these protections against channel distortions come with a cost of the data rate. More robustness leads to fewer modulated subcarriers K for carrying data. Moreover, different subcarrier modulation types and different channel coding rates in DRM also offer tradeoffs between data rate and robustness to channel distortion. The adaptivity of different transmission modes to different channel conditions in the DRM and the DAB system can give some indications on how to make an adaptive OFDM system for Cognitive Radio. The OFDM based physical layer is also used for digital video transmission standards such as the Digital Video Broadcasting (DVB) [24].

In addition to these radio systems, the OFDM technique has also been used for broadband wired applications like the asynchronous digital subscriber loop (ADSL) [3]. In the foreseeable future, the IEEE 802.16 standard[2](also known as WiMAX or Wireless MAN) may replace the traditional wired broadband access in our homes. The physical layer of the IEEE 802.16 standard adopts both OFDM and Orthogonal Frequency Division Multiple Access (OFDMA), a multiuser version OFDM system. The OFDMA mode enables a WiMAX base station (BS) to support multiple users at the same time. The OFDMA system for the WiMAX also offers multiple FFT sizes, cyclic prefix sizes, and pilot allocation schemes to make the system adaptive to various channel conditions and communication scenarios. Furthermore, OFDMA has already been proposed as the physical transmission scheme for Cognitive Radio based IEEE 802.22 standard. In conclusion, OFDM based transmission will be a strong candidate for future generation communication systems including Cognitive Radio.

3.2.3 The AAF Baseline OFDM System

OFDM has been considered as the physical layer transmission scheme for the emergency network in our AAF project. In [40], Hoeksema et al. proposed a set of parameters for the AAF baseline OFDM system. Any adaptivity and cognition will be added on top of the baseline system. The AAF OFDM system targets the spectrum from 400 MHz to 1000 MHz, which is similar to the band of interest for the IEEE 802.22 standard.

The AAF Parameter Set

By considering the typical channel conditions in the operating frequency band (400-1000MHz) and the data rate requirement of the system, a set of basic OFDM parameters for the AAF system was proposed in [40] (see Table 3.4 and Table 3.5). The system can occupy 5MHz or 1MHz bandwidth B. Since the typical doppler spread in the considered frequency range is less than 200Hz, the carrier spacing Δf is chosen as 10kHz which is much larger than the doppler spread. Therefore, the number of subcarriers are 512 and 128 for 5MHz and 1MHz bandwidth systems respectively. For both systems, the ratio between the useful time and the guard time $T_u : T_g$ can be chosen from 4:1, 8:1, 16:1 and 32:1 to give levels of robustness to different delay spreads.

[2] IEEE 802.16 Standard: `http://wirelessman.org/`

Table 3.4: OFDM parameters: sample frequency and symbol duration [40]

$B = f_s$	N	Δf	T_u
$[MHz]$		$[kHz]$	$[\mu s]$
1.28	128	10	100
5.12	512	10	100

Table 3.5: OFDM parameters: guard time and symbol duration [40]

$f_s = 1.28$		$f_s = 5.12$				
$[MHz]$		$[MHz]$				
N	N_g	N	N_g	T_u: T_g	T_g $[\mu s]$	T_s $[\mu s]$
128	32	512	128	4 : 1	25	125
128	16	512	64	8 : 1	12.5	112.5
128	8	512	32	16 : 1	6.7	106.7
128	4	512	16	32 : 1	3.4	103.4

Simulation Verification

To verify the parameters proposed in [40], a simulation has been conducted for the 5MHz bandwidth system by applying the typical channel model in the IEEE 802.22 standard [84]. The channel model for the IEEE 802.22 standard can be considered in the AAF context because the operating frequency and environments of both systems are similar. Two typical channels are considered: a channel with relatively small delay spread typically in rural areas (Channel 1 in figure 3.4) and a channel with large delay spread and attenuations typically for hilly terrains (Channel 2 in figure 3.4) [84]. We assume the coherence time of the channel is longer than one OFDM frame which consists of several OFDM symbols, thus the channel delay profile is time invariant during one OFDM frame. Therefore, the channel delay profile can be approximated as a Finite Impulse Response (FIR) filter. In the simulation, we assume the OFDM system is perfectly synchronized both in time and frequency. For simplicity, frequency domain equalization is employed by inverting the subchannel fadings at the receiver to compensate for the channel impairments. Moreover, no channel coding scheme is considered in the simulation. To evaluate the performance of communication systems, the Bit Error Rate (BER) is often plotted against the Signal-to-Noise Ratio (SNR). The SNR is defined as the ratio of the desired signal power to the

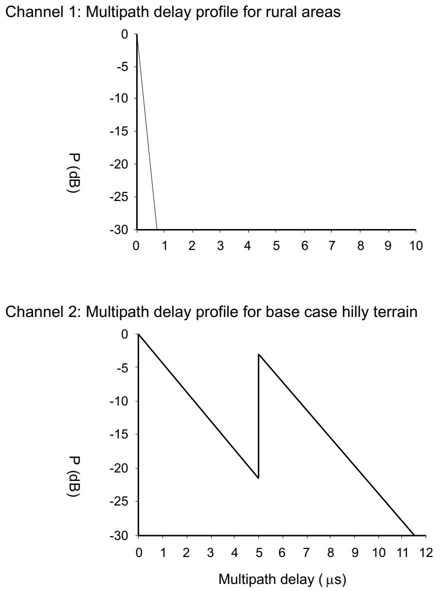

Figure 3.4: Typical channels for the IEEE 802.22 [84]

noise power. The SNR is an indication of signal quality in the transmission link between the transmitter and receiver. The BER is defined as the ratio of the number of bits incorrectly received to the total number of bits. The BER serves as an indication of reliability of the system.

In the first simulation, the 5MHz bandwidth system with the smallest number of guard samples $N_g = 16$ (see Table 3.5) is operated in Channel 1. QPSK, 16QAM and 64QAM are used as the subcarrier modulation schemes. Figure 3.5 shows the BER performance of the different modulation schemes under Channel 1, which is compared with the QPSK BER performance in an Additive Gaussian White Noise (AWGN) channel. From figure 3.5, we see that under a good channel such as Channel 1, even the smallest number of guard samples ($N_g = 16$) in the parameter set is sufficient to offer a relatively good performance.

In the second simulation, the 5MHz bandwidth system is operated in Channel 2. 16QAM is used as the subcarrier modulation scheme. Two different numbers of guard samples, $N_g = 128$ and $N_g = 16$, are considered. The $N_g = 16$ mode can only offer a guard time of 3.4 μs which is much less than the delay spread of Channel 2. Therefore, the unresolvable ISI in the

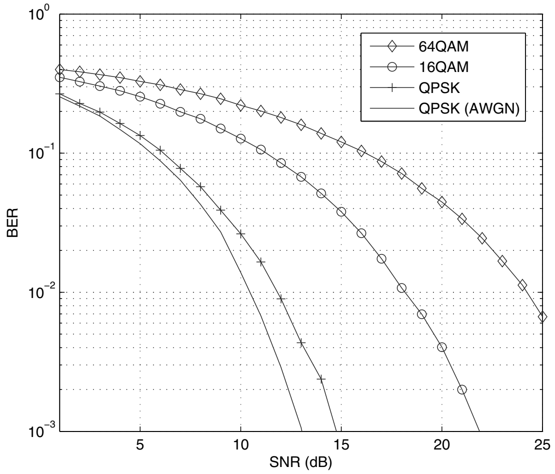

Figure 3.5: The BER performance of the AAF OFDM system under Channel 1 in figure 3.4

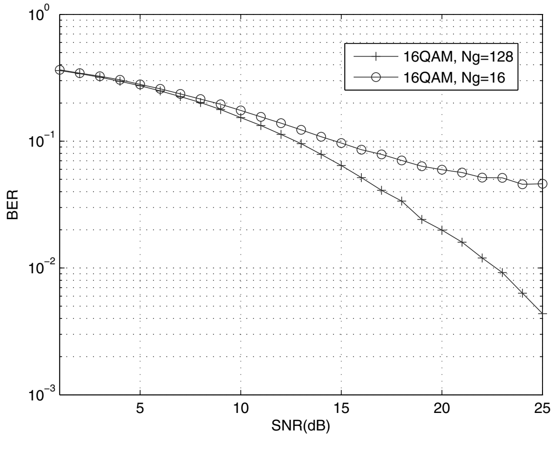

Figure 3.6: The BER performance of the AAF OFDM system under Channel 2 in figure 3.4

$N_g = 16$ case results in an error floor shown in figure 3.6. However, in the $N_g = 128$ case, the ISI can be resolved to a large extend as indicated by the performance difference. Therefore, the indication is that, when encountering a very bad channel scenario such as Channel 2, the system should switch to a more robust transmission mode.

From the simulation results, we can conclude that the choices of the parameters for the AAF system are valid under realistic channel models. Even in a very bad channel, the parameter set offers options for adaptation in order to enhance the performance.

3.3 OFDM for Cognitive Radio

Like normal OFDM systems, OFDM based Cognitive Radio systems have advantages such as high data rates, robustness to multipath fading and easy implementation using an FFT. In the context of Cognitive Radio, an OFDM system offers several additional benefits. First, subcarriers can be switched off (not to load data) to avoid interference to licensed users. This idea is also known as spectrum pooling [103]. Second, by adaptively loading subcarriers with different modulation schemes or transmit power levels the OFDM based Cognitive Radio can optimally use the segmented spectrum. Third, as an important spectrum analysis tool, the FFT module in OFDM can be reused for spectrum sensing. Moreover, as indicated in section 3.2, the OFDM system is very agile to adapt to different channel conditions by adapting its parameters. Furthermore, OFDMA as a multi-user version OFDM, can be applied to a multi-user Cognitive Radio system, for example the IEEE 802.22 system.

Besides the common challenges for OFDM such as its sensitivity to frequency offset, sensitivity to intercarrier interference and high peak to average power ratio, there are several problems which have to be solved for the OFDM based Cognitive Radio. First of all, the power leakage of subcarriers to adjacent subbands is a potential interference to the licensed system. Figure 3.7 shows an example of the subcarrier leakage problem. Due to the rectangular FFT window in the time domain, the shape of subcarrier appears to be a *sinc* function in the frequency domain [60]. The orthogonality enables these *sinc* shaped subcarriers to be overlapped with each other. Suppose one subcarrier as indicated by dotted line in figure 3.7 switches off to vacate the corresponding subband for licensed users. However, still considerable power from adjacent subcarriers is leaked into the vacated band thus causing interference to licensed users. Spectrum shaping and interfer-

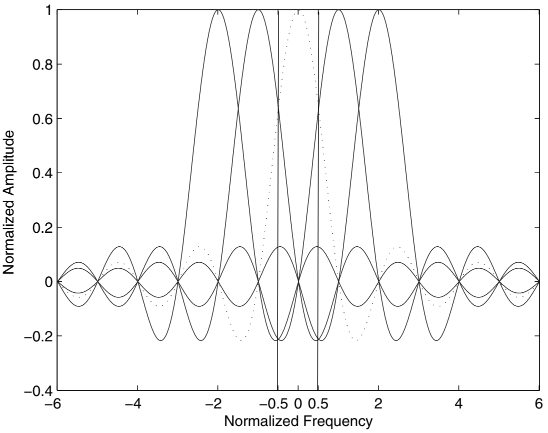

Figure 3.7: An example of subcarrier leakage in OFDM

ence cancellation are major issues for OFDM based Cognitive Radio. A lot of recent research effort on OFDM based Cognitive Radio is targeting these issues [104, 5].

Another concern for OFDM based Cognitive Radio is how to optimally use the segmented spectrum. As a solution, adaptive loading is proposed in conjunction with subcarrier deactivation for OFDM based Cognitive Radio [112, 8, 109]. Moreover, OFDM signal identification and transmission parameter estimation are also important for self-identification purposes [110]. Self-identification helps Cognitive Radio nodes establish their own network. From an implementation point of view, the deactivation of subcarriers offers opportunities to explore efficient algorithms in the transceiver design. Since the FFT is the most computationally complex part in OFDM, to develop more efficient FFT algorithms in this context is very beneficial [114, 66].

This section is dedicated to two aspects of OFDM based Cognitive Radio, especially in two aspects: spectrum pooling and adaptive loading. Finally, the conceived OFDM based Cognitive Radio system architecture used within this thesis is proposed.

3.3.1 Spectrum Pooling

The basic idea of spectrum pooling [103] is to merge spectral ranges from different spectrum owners (military, trunked radio etc.) into a common pool, where users may temporarily rent spectral resources during idle periods of licensed user. In this way, spectrum utilization can be improved by enabling

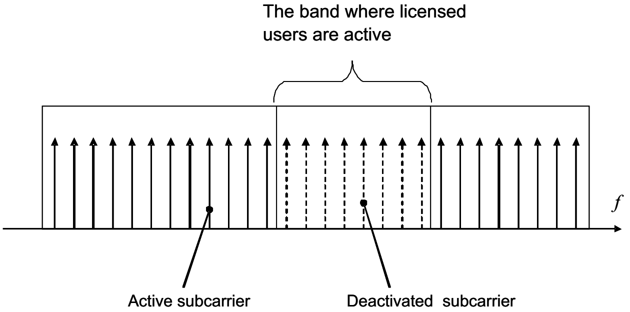

Figure 3.8: A schematic example of an OFDM based spectrum pool

the secondary utilization of already licensed frequency bands. The licensed system does not need to be changed. It behaves as if there was no secondary system present in the frequency range. The rental (secondary) system is required to have the cognition to identify the licensed system and adapt its transmission on a noninterference basis. In a spectrum pooling system, measurement information has to be collected by secondary users during a detection cycle to obtain the spectrum usage information. A mechanism has to be implemented to manage the spectrum usage information and make decisions on granting portions of the spectrum to the rental users (secondary users). A common control channel has to be established to disseminate the spectrum usage information within the rental network. From the physical layer perspective, the most challenging problem is how to reduce out-of-band radiation to enable successful coexistence.

OFDM is proposed as the physical layer transmission scheme for the secondary system in the context of spectrum pooling [103]. We refer to this OFDM based secondary system as OFDM based Cognitive Radio. The coexistence with the licensed system is realized by deactivating the subcarriers which causes interference to the active licensed users as shown in figure 3.8. The deactivation is done by loading zeros on the corresponding IFFT inputs. The deactivation information is regularly updated according to the spectrum usage information.

Due to the spectrum shape of OFDM, the out of band radiation of the secondary system may cause potential interference to the licensed system. On the other hand, the interference from the licensed system also degrades the performance of the secondary system. Mutual interference is the biggest challenge for the spectrum pooling system. This challenge has been recognized in [103] and a detailed numerical analysis on mutual interference in OFDM based spectrum pooling system has been given in [104]. An el-

Table 3.6: Relative interference power from RS to LS and vice versa [104]

n	1	2	3	4
$P_{R\rightarrow L}(n)$ [%]	4.922	0.848	0.483	0.325
$P_{L\rightarrow R}(n)$ [%]	0.913	0.107	0.079	0.063

liptically filtered white noise process was assumed as license signal. The interference of the OFDM based rental system (RS) to the licensed system (LS) is mainly due to the power spectrum leakage of the sinc shaped OFDM signal. The interference of the licensed system to the rental system is introduced by the FFT processing on the rectangular windowed time domain samples [104]. The relative interference power in one subband (subcarrier) as a percentage of the total transmit power in one subcarrier is derived in [104] (see Table 3.6). $P_{R\rightarrow L}(n)$ is the relative interference of the rental system to the licensed system while $P_{L\rightarrow R}(n)$ denotes the interference of the licensed system to the rental system. n is the distance between the considered subcarrier and adjacent subcarrriers. $n = 1$ indicates the immediate neighboring subcarriers and $n = 2$ indicates the second adjacent subcarriers and so on. The information from Table 3.6 indicates that the mutual interference in the OFDM based spectrum pooling system is quite considerable, particularly the interference from the immediate neighboring subcarriers. Moreover, the interference of the rental system to the licensed system is more dominant. Therefore, the pure OFDM system without any modification can hardly fulfil the coexistence requirements.

Two counter measures have been proposed to mitigate the inference to the licensed system by Weiss et al [104], namely windowing and deactivation of adjacent subcarriers. In the windowing method, a non rectangular window is applied in the time domain to make the amplitude go smoothly to zero at the symbol boundaries to obtain a sharp frequency response. To guarantee orthogonality, extra time domain samples have to be added for the windowed OFDM. In [104], a raised cosine window with the roll off factor β is considered. The results indicate that the windowing method is far from satisfactory. For example the "$\beta = 0.5$ raised cosine window" gives less than 6dB improvement in sideband attenuation compared to the rectangular window. However, the price paid for the improvement is extending the time domain samples by a factor of $(1 + \beta)$. In the $\beta = 0.5$ case, the OFDM symbol has to be extended with half the number of samples of the original symbol. As a result, the bandwidth efficiency is significantly reduced. The other solution is to deactivate more subcarriers adjacent to the licensed band. As indicated in Table 3.6, the interference from the fist adjacent subcarrier is the most significant. Therefore, the deactivation of the first adjacent subcarrier delivers the largest benefit. However, the question remains whether it is enough to mitigate the interference by deactivating only a few adjacent subcarriers since the interference is an accumulation of all active subcarriers in the rental system. Moreover, the deactivation sacrifices bandwidth of the rental system.

Since no optimal solution has been proposed to solve the mutual interference problem especially the interference to the licensed system, there is much space left for research. In [5], Brandes et al. proposed a method for further reducing sidelobes by assigning nonzero values to the deactivated subcarriers. The nonzero valued deactivated subcarriers, also refered to as the cancellation subcarriers, produce counter effects on the sidelobe. They claim as much as -60dB sideband rejection by applying the proposed method in conjunction with the raised cosine window in [5]. However, the drawback of the method is that the effort to generate these cancellation subcarriers is extremely high because a constraint optimization has to be done. Recently some researchers resort to other multicarrier transmission schemes to avoid the inherent spectrum leakage problem in OFDM [119, 2, 48]. As one of the contributions of this thesis, an oversampled filter bank multicarrier system with low out-of-band radiation is proposed in [119]. The details of the oversampled filter bank multicarrier approach will be presented in chapter 5.

3.3.2 Adaptive Loading

Since the channel, that an OFDM system is operating in, is usually frequency-selective, each subcarrier will have a different signal-to-noise ratio (SNR) within the corresponding subchannel. Thus, the system can be optimized by adapting the operating parameters of the subcarriers such as the choice of modulation scheme and/or power level to each subchannel. The process of tailoring subcarrier parameters is often referred to as adaptive loading including adaptive bit loading and power loading. The term bit loading defines a process of assigning a modulation scheme to each subcarrier, given a set of available modulation schemes, to achieve a performance objective while satisfying some prescribed constraint(s). The method of adjusting subcarrier power levels to change the subcarrier SNR and BER values is known as power loading or power allocation. Usually, power loading is performed in conjunction with bit loading. Adaptive loading has been developed and implemented for wirelined data transmission systems such as DSL modems [17]. Although

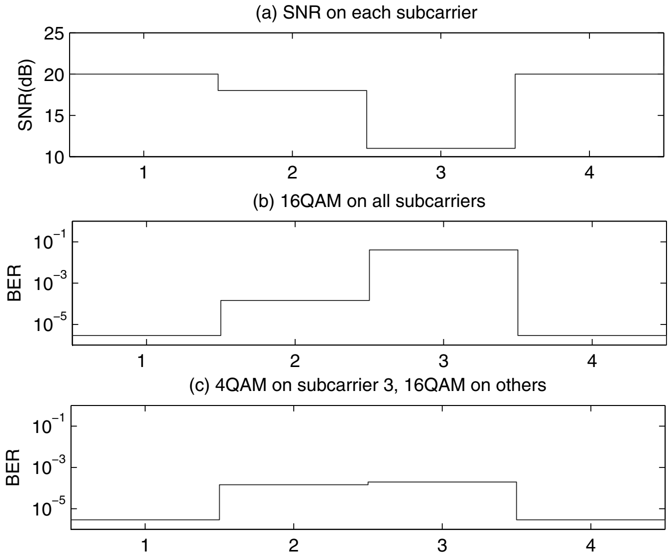

Figure 3.9: An example of bit loading for a 4 subcarrier system

adaptive loading is not employed by most wireless standards, it has been a hot research topic in wireless multicarrier systems [45, 108]. The rationale is that uniformly distributed operating parameters across all subcarriers do not exploit all the advantages offered by the multicarrier framework.

To illustrate how bit loading works and why it is beneficial for multicarrier systems, a simple example for a 4 subcarrier system is shown in figure 3.9. Due to different frequency selective fading, the SNR on the subcarriers differ as shown in figure 3.9 (a). The different SNRs result in different BERs on each subcarrier. If flat fading is assumed for each subcarrier, the Symbol Error Rate (SER) of M-QAM modulation on each subcarrier $P_{i,M}$ can be expressed as the function of the SNR γ_i of the corresponding sucbarrier in eq. 3.4 [65]:

$$P_{i,M}(\gamma_i) = 4(1 - \frac{1}{\sqrt{M_i}})Q(\frac{3\gamma_i}{M_i - 1})[1 - (1 - \frac{1}{\sqrt{M_i}})Q(\frac{3\gamma_i}{M_i - 1})] \qquad (3.4)$$

$Q()$ is the Q function, defined as:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-t^2}{2}} dt \qquad (3.5)$$

The BER can be approximated as:

$$P_i \approx P_{i,M} / \log_2(M_i) \qquad (3.6)$$

If 16QAM is applied for all subcarriers, the BER on each subcarrier is shown in figure 3.9 (b) by applying eq. 3.4 and eq. 3.6. The BER performance is more than 10^{-3} in subcarrier 3, which results in an overall BER degradation. If 4QAM is used for subcarrier 3 and 16QAM is applied to the others, the BER performance on subcarrier 3 is improved due to the robustness of lower order modulation (see figure 3.9 (c)). As a result, the overall BER performance is also improved to less than 10^{-3} in this case. However, the improvement leads to less throughput since less bits are transmitted in subcarrier 3. Loading schemes are not limited to the given example. For instance, the throughput can be improved by loading higher modulation modes on subcarriers which have high SNRs. Power loading can also be employed to boost bad subcarriers, like subcarrier 3, by transmitting more power in that subcarrier.

For OFDM based Cognitive Radio, adaptive loading will be even more beneficial. First, parameter adaption based on each subcarrier enables the system to be more adaptive to wireless channel conditions. Furthermore, since the spectrum resource in a spectrum pooling OFDM system is constrained, Cognitive Radio has to optimally use the spectrum to provide enough throughput and performance. For example, OFDM based Cognitive Radio has to deactivate some subcarriers to avoid emerging licensed systems and thus suffers throughput loss. If adaptive bit loading is applied, the throughput loss can be compensated by loading higher modulation modes to some active subcarriers with high SNRs. In such context, the spectrum pooling system combined with adaptive loading is foreseen [112, 8, 109].

However, adaptive loading also has several drawbacks. First of all, optimal adaptive loading demands perfect knowledge of the channel. Since perfect channel estimation can not be achieved in reality, the performance results of adaptive loading reported in literature may be overly optimistic. Secondly, the loading algorithm, usually an optimization procedure, increases the system complexity. Moreover, the loading algorithm has to be performed constantly depending on the channel variation. Therefore, to design a high performance yet low-complexity loading algorithm is crucial for a practical system. Thirdly, the loading information (e.g. modulation modes, transmit power levels) on each subcarrier has to be transmitted along with useful data. Such information induce a large overhead and will sacrifice valuable bandwidth.

3.3.3 Proposed System

A spectrum pooling system combined with adaptive loading, proposed in [112], is used as the system model for the OFDM based Cognitive Radio in this thesis. Mathematically, the adaptive loading can be formulated as an optimization problem.

In principle, an OFDM based Cognitive Radio can maximize the data rate under a certain power constraint. It is formulated as follows:

$$\begin{aligned}
&\textit{Max}\ \ R = \sum_{k=1}^{K} \frac{F_k}{K} \log_2(1 + \frac{h_k^2 p_k}{N_0 \frac{B}{K}}) \\
&\textit{Subject to:} \sum_{k=1}^{K} p_k \leq P_{total} \\
&F_k \in \{0, 1\} \text{ for all } k \\
&p_k = 0 \text{ for all } k \text{ which satisfy } F_k = 0
\end{aligned} \tag{3.7}$$

The data rate R is expressed as the form of the total system capacity which is the sum of the capacity of K subcarriers, where K denotes the total number of subcarriers. B and N_0 denote the bandwidth of Cognitive Radio and the average noise power respectively. Since fading on subcarrier h_k is assumed to be flat, h_k is a constant for each subcarrier. p_k is the power allocated to the corresponding subcarrier and P_{total} is the total power of the system. The deactivation information is given by the factor F_k, which takes either 1 or 0 to indicate the subcarrier k is either "on" or "off".

On the other hand, the system power minimization can also be applied under the constraint of a constant data rate. Similarly, it can be formulated as follows:

$$\begin{aligned}
&\textit{Min} \sum_{k=1}^{K} p_k = P_{total} \\
&\textit{Subject to: } R = \sum_{k=1}^{K} \frac{F_k}{K} \log_2(1 + \frac{h_k^2 p_k}{N_0 \frac{B}{K}}) \\
&F_k \in \{0, 1\} \text{ for all } k \\
&p_k = 0 \text{ for all } k \text{ which satisfy } F_k = 0
\end{aligned} \tag{3.8}$$

Furthermore, data rate optimization can be done under the system BER constraint and vice versa. These optimization principles serve as the basis to develop specific loading algorithms. The development of specific loading

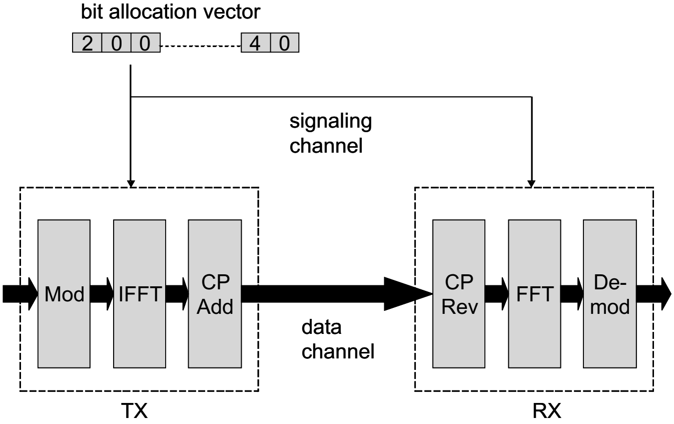

Figure 3.10: A block diagram of OFDM based Cognitive Radio

algorithms is beyond the scope of this thesis, but can be found in many papers [17, 28, 108]. However, the existing loading algorithms for pure OFDM systems still have to be shaped for Cognitive Radio, for example taking subcarrier deactivation into consideration. Moreover, loading algorithms for Cognitive Radio have to have low complexity and should incur a minimum overhead.

A block diagram of the proposed OFDM based Cognitive Radio system is presented in Figure 3.10. A bit allocation vector indicates how many bits are loaded on each subcarrier. The number of bits corresponds to the different modulation types used for each subcarrier. For example, one QPSK symbol consists of 2 bits and one 16QAM symbol contains 4 bits. The bit allocation vector is determined using the spectrum occupancy information from spectrum sensing and the SNR of the subchannels. A specific loading algorithm is responsible for generating the bit allocation vector. The bit allocation vector is disseminated via a signaling channel (or common control channel) such that both transmitter and receiver have the same information. We assume the bit allocation vector does not change frequently, for instance it remains constant during one OFDM frame. The basic idea is to take advantage of good subcarriers by loading more bits and loading zeros to (deactivate) subcarriers which cause interference to the licensed system or lead to poor transmissions. From a computation point of view, the difference between a pure OFDM system and the OFDM based Cognitive Radio system is that there could be a large number of zero inputs/outputs for the IFFT/FFT when a large part of the spectrum is not available to Cognitive Radio or there are many bad channels. In this case, the normal radix-2

IFFT/FFT will be inefficient due to the wasted operations on zeros.

3.4 TTL Modelling for OFDM based Cognitive Radio

The TTL approach as introduced in section 2.5 is used to model an adaptive OFDM system for Cognitive Radio [113, 116]. This TTL model gives a task graph based implementation which serves as a system-level description of the Cognitive Radio application. A task has processor specific implementations and these implementations are separated from inter-task communications. As a result, the TTL model will facilitate the mapping on the MPSoC platform.

3.4.1 General Approach

In a multiprocessor environment, a complete application description as a single sequential piece of code is difficult to implement since different parts of the application may need to run on different processors. Therefore, an application is often partitioned into tasks first, before mapping these onto a multiprocessor system. Streaming DSP applications are often quite naturally modelled as task graphs. As mentioned in section 2.5, the TTL model is based on task graphs. Creating a task graph of an application is the first step in the TTL method.

An OFDM system can be modelled as a chain of processing tasks. A non-adaptive OFDM system has a set of fixed parameters such as the FFT size, the guard time and modulation modes. As a result, the tasks are static and do not require reconfigurations. This is the case for a standard OFDM system like HiperLAN/2. However, in Cognitive Radio, an adaptive OFDM system is required. We consider such a system as a parameterizable OFDM processing chain where the tasks can be dynamically reconfigured. The TTL model introduced in [43] to model dynamic applications can be used to model such an adaptive OFDM system. An entity, called configuration manager (CM), is responsible for controlling and configuring/reconfiguring the parameterizable OFDM processing chain as shown in figure 3.11. Table 3.7 gives an example of parameters which can be reconfigured for the major OFDM processing tasks, but they are not limited by this table and can be extended to add more flexibility to the system. These reconfigurable parameters enable Cognitive Radio to adapt to the changing environment and on the other hand impose different processing requirements for the platform.

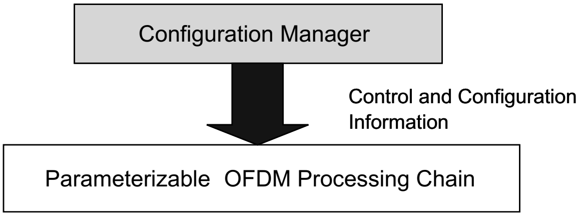
Configuration Manager
Control and Configuration Information
Parameterizable OFDM Processing Chain

Figure 3.11: The TTL model for adaptive OFDM for Cognitive Radio

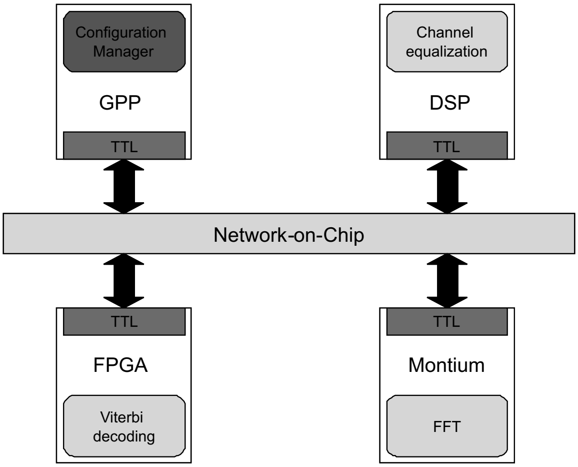

Figure 3.12: The implementation of adaptive OFDM on MPSoC with the TTL interface

Once the system-level description of the application in the TTL framework has been implemented (e.g. C/C++), it can be run on a PC to verify the functional behavior of the system and to generate profile information for computation and communication. The profile information helps the designer to identify where the system complexity is and to make mapping decisions on specific processors at an early design stage. Based on the TTL model, tasks can be implemented on specific processors on the platform. For example, a software task in TTL can be replaced with configuration data for the Montium or FPGA. TTL interfaces will be implemented on the targeted MPSoC. In figure 3.12, this is illustrated together with a possible distribution of several tasks in OFDM over different tiles.

Table 3.7: An example of a reconfigurable parameter set for adaptive OFDM system

Processing task	Parameters
Packet/Frame Detection and Synchronization	Frame size
	Frame header size
	Type of synchronization
Frequency Offset Estimation&Correction	Type of estimation
	Pilot for estimation
Channel Estimation and Equalization	Type of equalization
	Pilot for estimation
	Number of filter taps
Guard Time Removal	Symbol size
	Guard time size
FFT	FFT size
Phase Offset Tracking&Correction	Pilot for tracking
	Tracking method
De-mapping	Type of modulation
De-interleaving	De-interleaving method
	De-interleaving size
Channel Decoding	Type of coding
	Polynomial
	Code rate

3.4.2 The AAF Adaptive OFDM

In [116], we present the TTL model of the baseband processing of the AAF adaptive OFDM receiver. The reason to put the emphasis on the receiver is that the receiver is generally more computationally complex than the transmitter since it involves processing tasks to correct or compensate the distortion introduced by the channel. Modelling the transmitter in TTL can be done with the same approach. The basic parameters are taken from [40], where two sets of parameters are supported. Besides these basic parameters, the modulation type can be reconfigured to deactivate subcarriers.

The OFDM receiver for the AAF system is considered to have the following tasks for every OFDM symbol (see figure 3.13):

- **Guard time removal** is the task which removes the cyclic prefix and extracts the useful part for further processing.

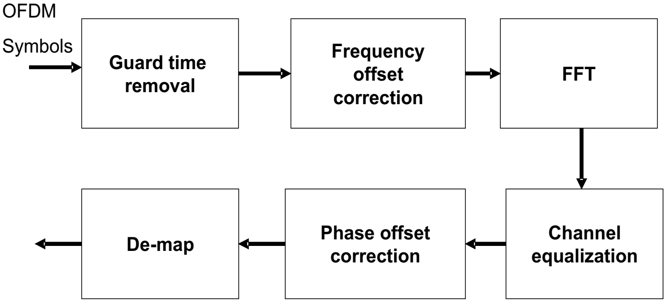

Figure 3.13: The OFDM receiver tasks

- **Frequency offset correction** corrects the frequency offset by multiplying all input samples with an inverse offset.
- **FFT** is the heart of the OFDM modulation.
- **Channel equalization** corrects the channel by multiplying data samples with the inverse of the channel gain.
- **Phase offset correction** corrects the frequency offset residual error by using the pilots in each OFDM symbol.
- **De-map** transforms the complex numbers to a bitstream according to their modulation schemes. Different modulation types are supported from 64QAM to BPSK and null subcarrier is also supported.

The reconfigurable parameters of these tasks are sent from the CM via control and configuration channel.

The TTL model is implemented in C/C++ and inter-task communications are function calls from the TTL library. The TTL implementation of the whole OFDM receiver in C/C++ can run on a Linux PC and therefore we can verify its functional correctness. The TTL run-time environment can generate high level profile information in terms of computation workload and communication workload. We made a computation workload analysis based on the TTL model. The instructions for the complex multiplication are annotated for analysis because they are the major contributors to the computational complexity. Table 3.8 shows the TTL generated computation workload of the 4 most computationally complex tasks in the AAF OFDM receiver. The computation workload is given as the number of complex multiplications for one OFDM symbol. The adaptive OFDM system can be reconfigured to support two sets of parameters for $N = 128$ and $N = 512$,

Table 3.8: Computation workload of the AAF OFDM receiver tasks

N	Frequency offset correction	FFT	Channel equalization	Phase offset correction
128	128	448	104	97
512	512	2304	416	385

Table 3.9: The estimated execution time in μs of the OFDM tasks on the Montium (run at 100MHz)

N	Frequency offset correction	FFT	Channel equalization	Phase offset correction
128	1.28	4.48	1.04	0.97
512	5.12	23.04	4.16	3.85

where N denotes the number of OFDM samples. Although the computation workload increases with the number of OFDM samples N, the FFT task is the most computationally complex task for both parameter sets. In this OFDM system, processing a 512-point FFT is the most computationally demanding task for the reconfigurable platform. If we consider that the worst case execution time (WCET) of the system should be less than the symbol duration 100μs (see Table 3.4), 2304 complex multiplications have to be done within 100μs. Therefore, the minimum processing capacity required by the adaptive parameterizable OFDM system is 23×10^6 complex multiplications per second.

If we consider mapping the AAF OFDM system on a Montium based platform, the profile information of the Montium for these tasks can be derived. A Montium processor can execute one complex multiplication instruction in one clock cycle. Assuming the Montium runs as high as 100MHz, the estimated execution time of the tasks for processing one OFDM symbol is given in Table 3.9. The power consumption of the Montium in 0.13μm technology is estimated at 0.577 mW/MHz [39]. We can use this figure to further estimate the energy consumption of each task for processing one OFDM symbol. For example, if we consider the $N = 512$ case, the energy consumption of the OFDM tasks for processing one OFDM symbol is illustrated in figure 3.14. The most energy hungry part is the FFT task which costs more than 1300 nJ for processing one OFDM symbol. Therefore to have an efficient FFT implementation is crucial to reduce the energy

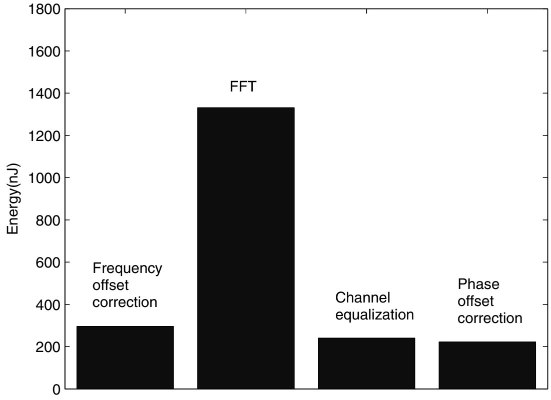

Figure 3.14: The energy consumption on the Montium for one OFDM symbol

consumption of the OFDM baseband processing.

We used a platform named BCVP [81] (Basic Concept Verification Platform) for verification purposes. This prototype multiprocessor platform is constructed with off-the-shelf components. It contains two ARM9 cores, a Xilinx FPGA, DRAM, two DDCs (Digital Down Converter), a Viterbi decoder and peripheral I/O. The Xilinx Virtex-II 8000 FPGA emulates three Montium tile processors connected by a circuit switched NoC (Network-on-Chip) [107]. A real-time operating system called BasOS [79] has been developed for the BCVP running on one of the ARM processors. It can implement the TTL type of interface functions for the inter-processor communications between the Montium and the ARM. The TTL model for the OFDM receiver described in this thesis has been ported to the BCVP platform with the support of the BasOS operating system [34]. The implementation[3]produced the results as expected. The task implementations in floating point precision in TTL are replaced by fixed point precision which are supported by the ARM and the Montium. Since the FFT is the most computationally intensive task, the processing of the FFT can be switched from the ARM to the Montium at run-time. The operating system enables the measurement of the execution time for each task processed on the BCVP. Although the measurements in [34] includes the overhead of the operating

[3] This has been part of the AAF Cognitive Radio demonstration at the DySPAN2008 conference (see figure 3.15).

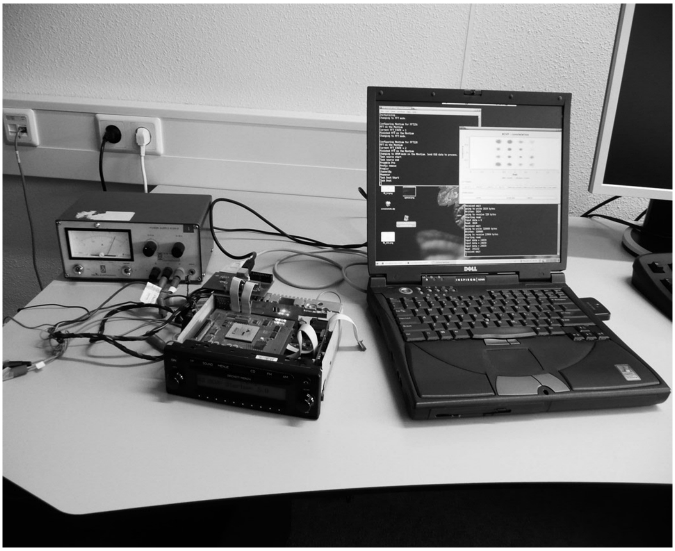

Figure 3.15: The AAF Cognitive Radio demonstration for the DySPAN2008 conference

system, it still confirms that the execution time of the FFT task is longer than any other task in the OFDM system if the overhead is assumed to have roughly the same effect on each task. Therefore, from a system point of view the FFT/IFFT task of the OFDM system is the bottleneck in terms of performance and energy consumption.

3.5 Chapter Summary

In this chapter, details regarding the basic physical layer transmission scheme of Cognitive Radio are discussed. OFDM, as a common multicarrier based transmission scheme, is considered for Cognitive Radio for the following reasons:

- it can provide agility in the frequency domain;
- it can use a segmented spectrum in an optimal way;
- it offers algorithm reusability for spectrum sensing;
- it offers high data rate and robustness to wireless channel impairments;
- it can be implemented by an FFT.

The advantages such as high data rate, robustness to a frequency selective channel and efficient FFT based implementation make OFDM a popular physical layer scheme in many practical systems. A brief overview of various OFDM applications has been given. We also discussed several basic parameters for designing an OFDM system. Finally, a baseline OFDM system in the context of the AAF project is proposed and verified by simulation under realistic channel models. Further adaptations will be added on top of the baseline system.

We focus on issues of OFDM in the context of Cognitive Radio. By switching off subcarriers, an OFDM based Cognitive Radio is able to avoid transmission within the band where a licensed system is active. This technique is also known as spectrum pooling. However, due to spectrum power leakage of OFDM, the power of the active subcarriers still causes potential interference to the licensed system. Moreover, OFDM based Cognitive Radio has to mitigate inference from a licensed system. As a result, the mutual interference problem becomes a major obstacle for OFDM based Cognitive Radio. Parameter adaption such as different modulation modes and power levels for each subcarrier, also known as adaptive loading, is very beneficial for OFDM based Cognitive Radio where the bandwidth resources are constrained because of subcarrier deactivation. Therefore, a combination of subcarrier deactivation and adaptive loading is proposed in section 3.3 as the OFDM based Cognitive Radio system model.

As the first step to map an OFDM based Cognitive Radio onto a reconfigurable platform, the adaptive OFDM system for Cognitive Radio is modelled using TTL. It is modelled as parameterizable OFDM processing tasks controlled and configured by a configuration manager. The system-level description of the application in the TTL model, implemented in software (e.g. C/C++), can run on a PC to verify the functional behavior of the system and to generate the profile information for computation and communication. The profile information provided by the TTL model for the AAF system gives a rough estimation on execution time and energy consumption of the processing tasks on a Montium based platform. From the profile information, we identified that the FFT/IFFT task is the most computationally intensive task. Therefore, an efficient implementation of the FFT is crucial to process an OFDM based Cognitive Radio on a SDR platform.

Chapter 4

Sparse FFT for OFDM Based Cognitive Radio

Due to the deactivation of subcarriers, there could be a large number of zero inputs/outputs for the IFFT/FFT in an OFDM based Cognitive Radio system. In this case, the normal radix-2 IFFT/FFT is inefficient due to wasted operations on zeros. Therefore, we propose a novel sparse FFT as an efficient option for OFDM based Cognitive Radio in this chapter[1]*. The proposed algorithm has been mapped onto the targeted reconfigurable platform. The complete mapping approach from the system-level modelling to the actual implementation on the Montium based platform will be discussed.*

[1] Parts of this chapter have been published in publications [QZ7] [QZ8] [QZ11].

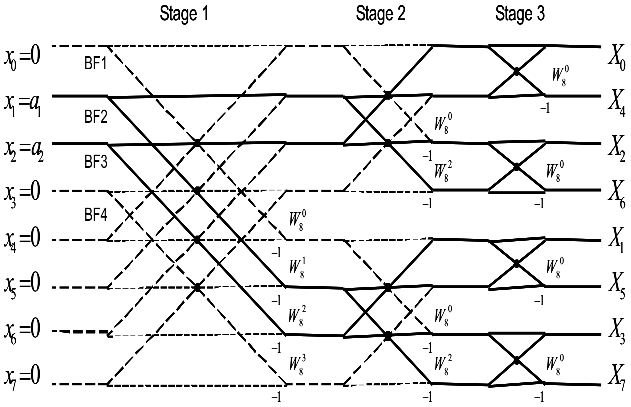

Figure 4.1: 8-point DIF radix-2 FFT

4.1 Motivation

From a system point of view, the FFT and IFFT are the critical parts of OFDM transceivers. They are also the most computationally intensive blocks in OFDM as confirmed by the TTL results in section 3.4. Therefore, an inefficient IFFT/FFT can considerably waste computational power of the overall system.

In an OFDM based Cognitive Radio system as shown in figure 3.10, there could be a large number of zero inputs/outputs for IFFT/FFT when a large part of the spectrum is not available to Cognitive Radio or there are many bad channels. If the number of zero values are relatively large, the standard FFT is no longer efficient due to the wasted operations on zero values. Such an example is given in figure 4.1. The computational structure of the 8-point Decimation-In-Frequency Radix-2 FFT in figure 4.1 shows a case where only two inputs are non-zero. All the operations indicated by the dashed line are actually wasted on the zero values. In the first stage, the first butterfly BF1 and the last butterfly BF4 are completely unnecessary and the other two butterflies (BF2 and BF3) are partial butterflies, which are less complex. Partial butterflies appear in the second stage as well. By pruning all those unnecessary operations, the computational power saving is considerable.

From a mathematical point of view, arrays or matrices where most of the elements have the same value (called the default value, usually 0) and only a few elements have a non-default value are *sparse*. It is beneficial and often necessary to take advantage of the sparse structure to reduce the operations of the standard algorithms.

In such a context, we propose a novel low complexity sparse FFT algorithm as an option for OFDM based Cognitive Radio in [114]. The algorithm will be presented in section 4.3. Mapping the proposed algorithm onto the targeted platform is considered in section 4.4.

4.2 Related Work

Efficient FFT algorithms for sparse inputs/outputs has been discussed in literature to a large extent. One well known algorithm is called *FFT Pruning.* As the name suggested, the basic idea is to prune off unnecessary calculations. In [51], a DIF (Decimation-In-Frequency) FFT pruning algorithm was suggested. Later the FFT pruning algorithm was extended to DIT (Decimation-In-Time) FFT [80] and to both input and output pruning [86]. In fact the common practice of all these pruning algorithms is to determine the index of the butterflies for calculations. The index is generated at runtime by conditional statements. However, the overhead to generate the index can be considerable due to the execution of conditional statements. The index shows irregularity because of the random position of zero inputs/outputs. This irregularity makes the hardware implementation of FFT pruning difficult. The idea of FFT pruning was applied to a multichannel OFDM system in [38], where a significant reduction of complexity was suggested. However, the application within OFDM in [38] assumes zero values with a considerable regularity. This is not true for Cognitive Radio where subcarriers are switched off at random positions based on the spectrum occupancy information and the subchannel condition.

In [85], Sorensen et al. proposed an efficient algorithm called *Transform Decomposition.* It is shown that transform decomposition is more efficient and flexible than FFT pruning. Transform decomposition can be seen as a modified Cooley-Tukey FFT [18] where the DFT is decomposed into two smaller DFTs. However, there are two limitations in the original work. Firstly, the discussions are restricted to the case where only the first L consecutive values are non-zero. Secondly, hardware implementation issues are not considered. Inspired by Sorensen's work, we proposed a novel sparse FFT [114] based on the modified Cooley-Tukey algorithm but overcame the aforementioned limitations ([85]).

In the context of Cognitive Radio, the authors in [66] also observed the fact that there could be many zero inputs/outputs for IFFT/FFT in the OFDM system. They proposed an FFT pruning method to design an efficient OFDM transceiver. However, like other FFT pruning methods the proposed method still has to determine the indices of butterflies that need to be calculated. Moreover, their approach is mainly software oriented and did not consider hardware issues. In a recent work [76], the authors proposed an efficient FFT for an OFDMA system based on the transform decomposition. However, they only considered the case that the number of nonzero values is a power-of-two.

4.3 Proposed Algorithm

The basic idea of the sparse FFT is to divide a large FFT into smaller size FFTs, which offers the opportunity to reduce computational complexity.

4.3.1 FFT Module

Below, a mathematical derivation is given. The DFT is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \ k = 0, 1, ..., N-1 \tag{4.1}$$

, where $W_N^{nk} = e^{\frac{-j2\pi nk}{N}}$. We consider the case where L outputs are nonzero. Let N be factorized as two integers N_1 and N_2, so $N = N_1 N_2$. To facilitate the implementation, we make the size of the FFT N, a power-of-two integer. We choose N_1 as the nearest power-of-two integer larger than L and as a factor of N, denoted as $N_1 = \lceil L \rceil_{pow2}$. Therefore, N, N_1 and N_2 are all power-of-two integers. The index n $(0 \leq n \leq N-1)$ can be written as:

$$\begin{gathered} n = N_2 n_1 + n_2 \\ n_1 = 0, 1, ..., N_1 - 1 \quad n_2 = 0, 1, ..., N_2 - 1 \end{gathered} \tag{4.2}$$

Substituting n in (4.1) with (4.2), the DFT can be rewritten as:

$$\begin{aligned} X(k) &= \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x(N_2 n_1 + n_2) W_N^{(N_2 n_1 + n_2)k} \\ &= \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x(N_2 n_1 + n_2) W_N^{N_2 n_1 k + n_2 k} \\ &= \sum_{n_2=0}^{N_2-1} [\sum_{n_1=0}^{N_1-1} x(N_2 n_1 + n_2) W_N^{N_2 n_1 k}] W_N^{n_2 k} \end{aligned} \tag{4.3}$$

We concentrate on the term $W_N^{N_2 n_1 k}$ within (4.3) and define $k = k'N_1 + \langle k \rangle_{N_1}$ where k' is an integer and $\langle \rangle_{N_1}$ denotes modulo N_1. Then:

$$\begin{aligned} W_N^{N_2 n_1 k} = W_N^{N_2 n_1 (k'N_1 + \langle k \rangle_{N_1})} &= W_N^{N_1 N_2 n_1 k' + N_2 n_1 \langle k \rangle_{N_1}} \\ = W_N^{N(n_1 k')} W_{N_1 N_2}^{N_2 n_1 \langle k \rangle_{N_1}} &= 1 * W_{N_1}^{n_1 \langle k \rangle_{N_1}} \end{aligned} \quad (4.4)$$

Then (4.3) can be written as:

$$X(k) = \sum_{n_2=0}^{N_2-1} [\sum_{n_1=0}^{N_1-1} x(N_2 n_1 + n_2) W_{N_1}^{n_1 \langle k \rangle_{N_1}}] W_N^{n_2 k} \quad (4.5)$$

We define $X_{n_2}(\langle k \rangle_{N_1})$ and $x_{n_2}(n_1)$ such that:

$$\begin{aligned} X_{n_2}(\langle k \rangle_{N_1}) &= \sum_{n_1=0}^{N_1-1} x(N_2 n_1 + n_2) W_{N_1}^{n_1 \langle k \rangle_{N_1}} \\ &= \sum_{n_1=0}^{N_1-1} x_{n_2}(n_1) W_{N_1}^{n_1 \langle k \rangle_{N_1}} \end{aligned} \quad (4.6)$$

So (4.3) can be written as:

$$X(k) = \sum_{n_2=0}^{N_2-1} X_{n_2}(\langle k \rangle_{N_1}) W_N^{n_2 k} \quad (4.7)$$

The original N-point DFT with L nonzero outputs is decomposed into two major parts: N_2 times an N_1-point DFT in (4.6) which can be implemented as N_2 times an N_1-point FFT, and the multiplications with twiddle factors and summation in (4.7). Because $X(k)$ only consists of L nonzero values, only L twiddle factors are multiplied with each $X_{n_2}(\langle k \rangle_{N_1})$ for $n_2 = 1, 2, ..., N_2$. This multiplication part results in a computational reduction.

To give a better understanding, we show a small example of a sparse FFT. Let us consider a 128-point FFT where only 7 outputs are nonzero ($L = 7$). We divide 128-point DFT into two smaller sized DFTs: 8 point DFT and 16 point DFT. N_1 is chosen as 8 because 8 is the nearest power-of-two integer larger than 7. $N_2 = 16$ because $N_1 = 8$ and $N = N_1 N_2 = 128$. Therefore, we have 16 times 8-point FFT and 105 ($(N_2 - 1)L = 15 \times 7 = 105$) multiplications with twiddle factors which only produce nonzero outputs. So in total we have $16 \times \frac{8}{2} \log_2 8 + 105 = 297$, while a 128 point FFT would take $\frac{128}{2} \log_2 128 = 448$ multiplications.

4.3.2 IFFT Module

For completeness, we also derive the sparse inverse FFT (IFFT). The IDFT is defined as:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-nk}\ n = 0, 1, ..., N-1 \tag{4.8}$$

where $W_N^{-nk} = e^{\frac{j2\pi nk}{N}}$. In the IFFT case, we consider the case where L *inputs* are nonzero corresponding to L active subcarriers. Let N be factorized as two integers N_1 and N_2, so $N = N_1N_2$. The choices of N, N_1 and N_2 are the same as for the sparse FFT. The index n can be written as:

$$\begin{gathered} n = N_2n_1 + n_2 \\ n_1 = 0, 1, ..., N_1 - 1 \quad n_2 = 0, 1, ..., N_2 - 1 \end{gathered} \tag{4.9}$$

Substituting n in (4.8) with (4.9), the IDFT can be rewritten as:

$$x(N_2n_1 + n_2) = \frac{1}{N}\sum_{k=0}^{N-1}[X(k)W_N^{-n_2k}]W_{N_1}^{-n_1k} \tag{4.10}$$

If we define:

$$X_{n_2}(k) = X(k)W_N^{-n_2k} \tag{4.11}$$

then eq. 4.10 can be written as:

$$x(N_2n_1 + n_2) = \frac{1}{N}\sum_{k=0}^{N-1} X_{n_2}(k)W_{N_1}^{-n_1k} \tag{4.12}$$

The index k can be written as:

$$\begin{gathered} k = N_1k_2 + k_1 \\ k_1 = 0, 1, ..., N_1 - 1 \quad k_2 = 0, 1, ..., N_2 - 1 \end{gathered} \tag{4.13}$$

Substituting eq. 4.13 into eq. 4.12, we get

$$x(N_2n_1 + n_2) = \frac{1}{N}\sum_{k_1=0}^{N_1-1}[\sum_{k_2=0}^{N_2-1} X_{n_2}(N_1k_2 + k_1)]W_{N_1}^{-n_1k_1} \tag{4.14}$$

From eq. 4.14, we can see that the sparse IFFT also consists of two parts: N_2 times an N_1-point IDFT which can be implemented as N_2 times an N_1-point IFFT and the multiplications with twiddle factors and recombinations of the multiplications in the term $\sum_{k_2=0}^{N_2-1} X_{n_2}(N_1k_2 + k_1)$. Similar to the sparse FFT, the computation reduction of sparse IFFT also comes from the multiplication part.

4.3.3 Complexity Analysis

Since the IFFT is just a symmetric reverse process of the FFT, the computational complexity is the same as for the FFT. We therefore only analyze the computational complexity of the sparse FFT. We use the number of complex multiplications as a measure of complexity. The number of complex multiplications in an N point sparse FFT with L nonzero outputs equals:

$$Mul_{sparse} = (N_2 - 1) * L + \frac{N}{2} \log_2 N_1 \tag{4.15}$$

The first term $(N_2 - 1) * L$ is the number of multiplications with $W_N^{n_2 k}(n_2 = 1, 2, ...N_2 - 1)$ and the second term $\frac{N}{2} \log_2 N_1$ is the number of multiplications needed for N_2 times N_1 point FFT. The number of complex multiplications in a N point radix-2 FFT is well known as $\frac{N}{2} \log_2 N$ [18]. Compared to the radix-2 FFT, the sparse FFT is more efficient for small L. For the sparse FFT, the complexity increases with the number of non zeros L and reaches the complexity of the normal radix-2 FFT when $L = \frac{N}{2}$. Since we set the constraints $N_1 = \lceil L \rceil_{pow2}$ and $N = N_1 N_2$, the sparse FFT will be calculated as a normal N point FFT when $L > \frac{N}{2}$. These constraints are important improvements to the algorithm in [85] in the sense that they facilitate hardware implementations by exploiting power-of-two integers. In [85], the discussions are restricted to the case when only the first L consecutive values are nonzero. Our proposed algorithm can be applied to any nonzero distribution. Assuming the size of the FFT as $N = 2048$, we compared the complexity of the sparse FFT and the radix-2 FFT in figure 4.2. The sparse FFT has less complexity than the radix-2 FFT when the nonzero ratio $\frac{L}{N} < 0.5$. The efficiency increases with the decrease of the nonzero ratio $\frac{L}{N}$, especially in the region where the nonzero ratio $\frac{L}{N} < 0.25$.

4.3.4 Discussions

In the context of OFDM based Cognitive Radio, the sparse FFT is an efficient option in cases where a large number of subcarriers have to be deactivated. However, to apply this algorithm to a real system we have to consider several issues:

- The gain of the algorithm with respect to the nonzero ratio $\frac{L}{N}$

  The nonzero ratio $\frac{L}{N}$ has to be considered for applying the sparse FFT within an OFDM system. The choice of the threshold depends on both the deactivation information and how much computation saving the

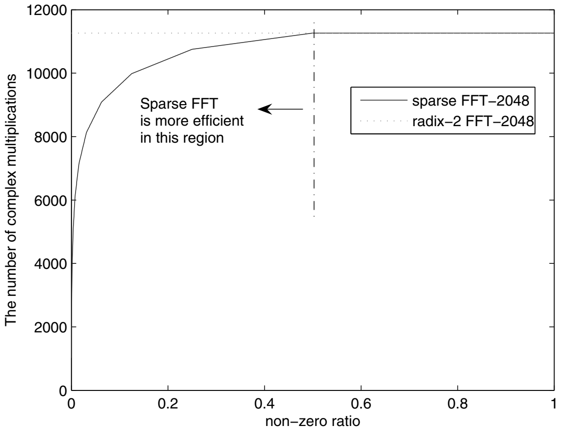

Figure 4.2: Complexity comparison of the sparse FFT

designer wants to achieve. If the deactivation information indicates that the number of active subcarriers is always more than half of the total subcarriers, there is no saving in using the sparse FFT.

- The reconfigurability

  In the OFDM based Cognitive Radio system, the subcarrier deactivation information is constantly updated depending on spectrum occupancy information and channel conditions. The deactivation pattern has a great impact on the system. The radix-2 FFT is used when the ratio of active subcarriers is larger than a threshold. While below the threshold, the system will be reconfigured to the sparse FFT. Therefore, an efficient dynamically reconfigurable sparse FFT module is desirable for the system.

- The update of deactivation information

  Another important concern for applying the algorithm is how fast the deactivation information is updated. If the deactivation pattern is rapidly changing, the benefit of the algorithm will diminish due to the overhead of reconfiguration.

Another interesting application of the sparse FFT is in the OFDMA system for Cognitive Radio based IEEE 802.22 standard. In OFDMA, multiuser data is spread across several non-contiguous bands and only a small portion of subcarriers (e.g. 128 out of 2048) are used by a single user. This

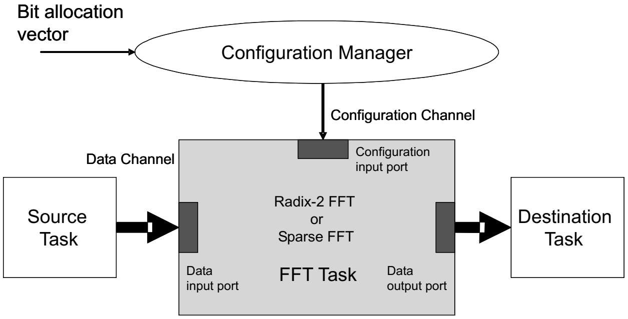

Figure 4.3: Task graph of reconfigurable sparse FFT for OFDM based Cognitive Radio

scenario is quite similar to the OFDM based spectrum pooling system. A recent publication [76] also observed this fact and proposed their efficient FFT solution for the OFDMA system.

4.4 Mapping onto the Reconfigurable Platform

As an effort to map an adaptive physical layer onto the reconfigurable platform, the proposed sparse FFT is mapped onto the targeted platform. This section will present the complete mapping approach from the system-level modelling to the actual implementation on the Montium based platform.

4.4.1 Sparse FFT in the TTL Model

The design flow starts from a system-level description of the algorithm in the TTL model. The reconfigurable sparse FFT is implemented in the TTL environment [115] to achieve the following goals: 1) to verify the sparse FFT at system level; 2) to obtain high level profile information in terms of computation workload and communication workload which assists making the implementation trade-offs on processors.

During the first step, we create the task graph (see figure 4.3) of the reconfigurable sparse FFT. The source task generates the input samples which are sent via the data channel to the FFT task. The destination task consumes the output samples from the FFT task. A configuration manager decides the type of FFT algorithm, depending on the number of non-zero values L in the bit allocation vector which indicates the subcarrier deactivation pattern (shown in figure 3.10). If $L < N/2$, the configuration manager will generate the configuration data for a sparse FFT since it is

```
Task Task_FFT
{initialization;
while(true)
 {local variables;
 //check the configuration updates
 tryAcquiredata(Task_FFT→config_inport)
 {//update L
 ttl_read(Task_FFT→config_inport, L);
 //read in configuration
 ttl_read(Task_FFT→config_inport, SFFT_CONFIG_DATA);
 }
 //read in data
 for(i=0; i<num_samples; i++)
 ttl_read(Task_FFT→data_inport, proc_buffer[i]);
 //sparse FFT or radix-2 FFT
 if (L<num_samples/2)
 {//sparse_FFT processing
 call sparse_FFT;
 }
 else
 {//radix-2 FFT
 call rad2_FFT;
 }
 //write out results
 for(i=0; i<num_samples; i++)
 ttl_write(Task_FFT→data_outport, proc_buffer[i]);
 }
}
```

Figure 4.4: Pseudo code of the TTL implementation of the FFT task

more efficient in this case. It indicates the FFT task to perform a sparse FFT and sends all the configuration data to the FFT task via the configuration channel. The configuration manager will go to standby until a new bit allocation vector arrives. Depending on L, the FFT task either performs a radix-2 FFT or a sparse FFT.

TTL functions are called from the TTL C/C++ library to create tasks, to define communication interfaces and to generate the task graph. At system level, the tasks are coded in C/C++, but in the platform implementation, the tasks can be implementations for a particular processor. We give a pseudo code example of the TTL implementation of the FFT task in figure 4.4 to show how reconfiguration is done for the FFT task. Before starting a new FFT, the FFT task checks for updates from the configuration channel. If a new configuration is generated by the configuration manager, the FFT task will read the configuration data from the configuration channel via the configuration input port. Then the FFT task reads samples from the source task. After reading the samples and the configuration data, the sparse FFT or radix-2 FFT procedure will be executed depending on L and N. After FFT processing, the results will be written out to the data channel via the data output port. Both synchronization and data transfer are done by TTL read and write functions. The procedures for sparse FFT and radix-2 FFT are software implementations in C/C++, but they can also be replaced by equivalent hardware implementations; for example, configurations for the Montium processor.

We applied the reconfigurable sparse FFT to an OFDM receiver for the AAF system based on the specification in Table 3.4. The bandwidth for the OFDM system is chosen as 5.12MHz, so there are 512 subcarriers in one OFDM symbol. A 512-point FFT is needed to process an OFDM symbol, where some subcarriers might be zero according to the bit allocation vector. An example of the sparse FFT reconfiguration for the given OFDM system is shown in figure 4.5. We denote all the zero output indexes of the FFT with 0 and nonzero indexes with 1. In Scenario 1, 420 of 512 indexes are nonzeros which means that most of the subcarriers can be used by Cognitive Radio. To avoid causing interference to a potential licensed user, Cognitive Radio has to switch off a certain number of subcarriers and has to re-assign the information to be transmitted to the available subcarriers. This corresponds to Scenario 2 where only 56 out of 512 indexes are nonzeros. From Scenario 1 to Scenario 2, the FFT task is reconfigured from a radix-2 FFT to a sparse FFT.

The high level TTL model has been implemented on a Linux PC. By running a set of test data, the functional correctness of the sparse FFT can be verified. The TTL run-time environment can generate high level profile information in terms of computation workload and communication workload. The computation workload is measured by counting the number of annotated instructions while the communication workload is measured by counting the number of tokens (data units) that are travelling through the TTL channels. Table 4.1 shows the computation workload of the FFT task in the two scenarios (see figure 4.5) generated by TTL. The reduction of computation in Scenario 2 is due to the sparse FFT which takes advantage of sparse data. Considering that the worst case execution time (WCET) for processing an OFDM symbol is $100\mu s$, we estimate the minimum required processing capacity for the Scenario 1 and Scenario 2 in Table 4.2. The profile information generated by the TTL run-time environment is independent of platforms. However, it can help to generate a platform dependent profile for specific implementations. By associating execution times with instruc-

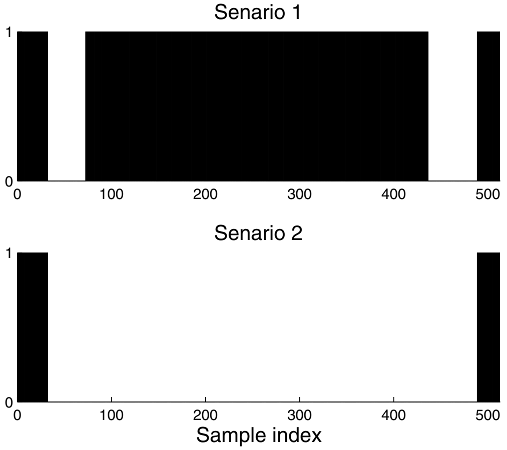

Figure 4.5: Example of reconfiguration of OFDM based Cognitive Radio

Table 4.1: Computation workload of the FFT task

	complex multiplication (instruction count)
Scenario 1	2304
Scenario 2	1928

tions, and by multiplying these execution times with the instruction counts, one can obtain a rough estimate of the total execution time of a task on a certain processor. Considering the Montium for the FFT task, the Montium can execute one complex multiplication instruction in one clock cycle. From Table 4.2, we find that the Montium has to run at at least 23MHz for Scenario 1 and 19MHz for Scenario 2. In other words, the sparse FFT will save 16% processing capacity. A reconfiguration only takes place when the bit allocation vector has been updated. We expect the bit allocation vector not to change very often: it will at least be constant during several OFDM frames. Therefore the reconfiguration overhead is relatively small compared to the saving of computations. We compare the computation workload of a 512-point sparse FFT for various L with different zero distributions using the TTL implementation. Figure 4.6 shows that the computation workload increases with the number of non-zero L.

From this design case, the TTL approach has given the following advantages.

Table 4.2: Minimum processing requirements

	complex multiplications (per second)
Scenario 1	23×10^6
Scenario 2	19×10^6

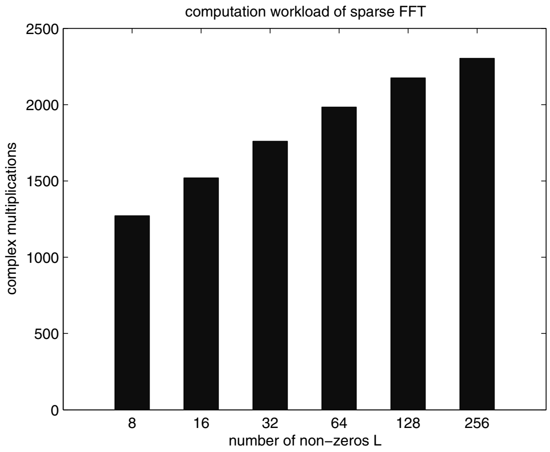

Figure 4.6: Computation workload of sparse FFT for 512 samples

- The functional correctness of the proposed algorithm could be verified at system level.
- The TTL programming model facilitates the application partitioning and the task based implementation on MPSoCs.
- The profile information given by the TTL run-time environment is used for the computation and communication complexity analysis at system level. Design trade-offs can be made at an early design stage.
- TTL is suitable for modelling a reconfigurable application.
- The TTL code is portable to the reconfigurable platform.

4.4.2 Dynamically Reconfigurable FFT on the Montium

Since the Montium is a key element in the proposed platform, mapping the proposed sparse FFT onto the Montium is the next design task.

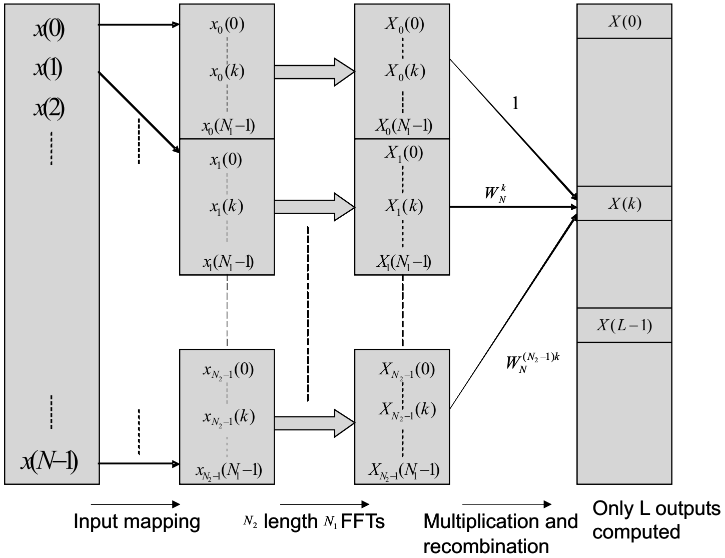

Figure 4.7: Computational structure of the sparse FFT

General Considerations

Before going into the details of the implementation on the Montium, we elaborate some features of the sparse FFT which are generally applicable to any hardware implementation.

First of all, the proposed algorithm shows considerable regularity which facilitates its hardware implementation. Figure 4.7 shows the computational structure of the sparse FFT. Basically the computation can be divided into two stages: FFTs and multiplications with recombination. Before the FFT computation, the input samples are mapped to N_2 memory blocks according to the index relation in eq. 4.2. Then the N_1-point radix-2 FFT is performed on each of the N_2 memory blocks. The N_2 results of the $k\ mod\ N_1$ memory position in each memory block are multiplied with twiddle factors and recombined to produce the output X_k. A regularity can also be found in the memory addressing: a constant hop from the same position in one block to another (e.g. from $x_i(0)$ to $x_{i+1}(0)$). Because N, N_1 and N_2 are all power-of-two integers, we can use the most significant bits to indicate the block address. This block based addressing is done by changing the most significant bits. For example in figure 4.8, suppose 32 memory positions are divided into 4 blocks with 8 memory positions in each block. The first two bits are used as block address. If the address changes from position 1 in

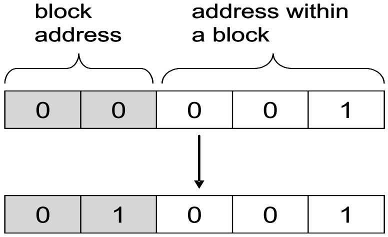

Figure 4.8: A block address example

the first block to position 1 in the second block, we increase the word which consists of two most significant bits from 00 to 01. The overall address is changed from 00 001 to 01 001.

Secondly, radix-2 FFT is part of the sparse FFT, therefore there is a reusability of the radix-2 FFT for the sparse FFT. For a software implementation, the software module of a radix-2 FFT can be reused in the sparse FFT. From a hardware architecture point of view, the butterfly architecture of the radix-2 FFT can be reused for the sparse FFT.

Thirdly, the computational structure of the sparse IFFT resembles the structure of the sparse FFT. The only difference is the order of computation. From eq. 4.14, we can see that the sparse IFFT first does the multiplication and recombination followed by the radix-2 IFFT stage.

Furthermore, the algorithm is scalable. The complexity of the algorithm depends on the nonzero ratio.

The Montium Implementation

OFDM based Cognitive Radio is expected to switch between sparse FFT and radix-2 FFT according to the subcarrier deactivation information. Moreover, it may use a variable sized radix-2 FFT for different parameter sets. Therefore, the dynamic reconfigurable FFT module implemented on the Montium should enable two types of reconfiguration: 1) the reconfiguration of the FFT size; 2) the reconfiguration between sparse FFT and radix-2 FFT.

- The basic idea to make a size-reconfigurable FFT is to reuse the computation structure and the twiddle factors of larger FFTs for smaller FFTs. An example is shown in figure 4.9 where a 4 point FFT can be realized from the structure of an 8 point FFT by skipping the last stage butterfly operations. The same rule applies for realizing any smaller size radix-2 FFTs from a larger radix-2 FFT. In the implementation,

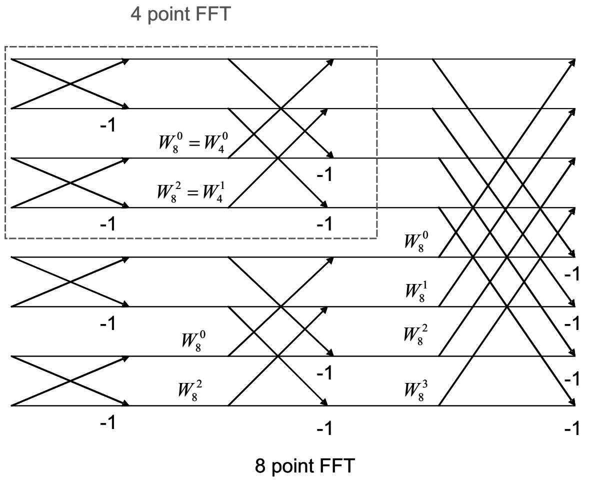

Figure 4.9: An example of reconfigurable radix-2 FFT

we load the configuration of the largest FFT needed into the Montium sequencer (see section 2.4 for details) during the initialization stage. Run-time reconfiguration is achieved by altering the configuration memory. Each entrance in this memory defines an instruction and contains a control part which is used to implement a sequencing state machine. An FFT algorithm consists of a number of stages depending on the size of the FFT. Each stage consists of several butterfly operations. One butterfly operation can be computed in one clock cycle using four Montium ALUs [39]. The butterfly computations are done sequentially on a stage by stage basis. Since the butterfly operation has a repeating pattern, it can be implemented in a loop. By reducing the loop counter values in the sequencer instructions and only using the stages needed, the large FFT can be reconfigured to a small FFT. Switching back to a large FFT can also be done by adjusting the loop counters and using more stages in the sequencer.

To give a better understanding of how the reconfiguration is done on the Montium, we show an example of a simplified sequencer program in Table 4.3 for a reconfigurable FFT changing from FFT-16 to FFT-8. The numbers below the AGU and ALU column represent indices of specific Address Generation Unit (AGU) or ALU instructions. The contents of these AGU and ALU instructions are trivial in this discussion, thus omitted. The FFT-16 has 4 stages with 8 butterfly operations on each stage while the FFT-8 has 3 stages with 4 butterfly operations. In order to change this 16 point FFT sequencer program to an 8 point FFT, the loop counter values have to be adjusted from 8 to 4 and a stage has to be skipped by adjusting the jump target in the sequencer instruction. The changes are indicated by an asterisk in Table 4.3. These changes in the sequencer are actually the reconfiguration data needed for the Montium to switch from an FFT-16 to an FFT-8.

Table 4.3: A simplified sequencer program for a reconfigurable FFT changing from a 16 point to an 8 point FFT

Seq PC	ALU	AGU	jump	loop	description
[0]	[0]	[3]	[1]	1	Start of stage 1
[1]	[1]	[4]	[2]	8 (4)*	
[2]	[1]	[4]	[3]	1	Start of stage 2
[3]	[0]	[3]	[4] ([7])*	8 (4)*	
[4]	[0]	[3]	[5]	1	Start of stage 3
[5]	[1]	[4]	[6]	8	
[6]	[1]	[4]	[7]	1	Start of stage 4
[7]	[0]	[3]	[8]	8 (4)*	

- During the input mapping and the recombination stage, the memory addressing is constantly hopping from a position in one block to the same position in the next block as indicated in figure 4.7 (e.g. from $x_0(0)$ to $x_1(0)$). Each memory of the Montium has an Address Generation Unit (AGU) which can generate the required addressing pattern. This practice has been used in [73] to map a non power-of-two FFT onto the Montium. Recall the computational structure in figure 4.7, the radix-2 FFT configuration can be re-used for each memory block in the first part. In the second part, where multiplications with twiddle factors and recombination are done, the 5th ALU is used to calculate the indices to be multiplied and is used as a memory address for the AGU [100]. These 3 extra clock cycles for address generation per nonzero value leads to an efficiency bottleneck of the sparse FFT on the Montium as will be shown in the result. One solution to this bottleneck is to move the task of this address generation from the Montium to the GPP on the platform. The generated addresses can be sent to the Montium as part of configuration data, however, this will increase the size of the configuration data which may exceed the

Montium configuration space.

Result

Figure 4.10 shows the number of clock cycles of the sparse FFT compared to the radix-2 FFT for 512 samples on the Montium as a function of the number of non-zero values. Due to the address generation overhead mentioned earlier, the final sparse FFT implementation on the Montium turns out to be not as efficient as the theoretical prediction. As shown in figure 4.10, the complexity of the sparse FFT exceeds the radix-2 FFT when there are more than 64 non-zero outputs. However, it is still more efficient than the radix-2 FFT for example when a large number of subcarriers are switched off (e.g. 480 out of 512). An additional benefit of the sparse FFT is that only the non-zero outputs are sent through the on-chip communication architecture. Therefore it results in less communication cost which is considered to be a major bottleneck for smaller feature sized SoCs in the future. The result in figure 4.10 also differs from the TTL results because TTL does not consider the implementation details. However, TTL results give a reference for computational complexity analysis. They are especially useful for large and complex systems.

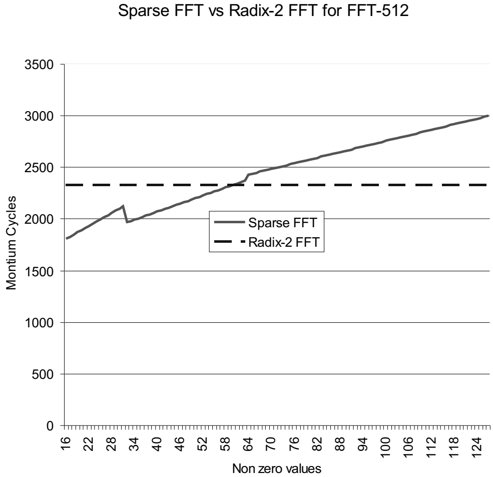

Figure 4.10: The performance of the sparse FFT vs radix-2 FFT for FFT-512 on the Montium

Table 4.4 shows the costs of the reconfiguration for both the radix-2 FFT and a sparse FFT (less than 64 non-zeros) in terms of bytes. When using reconfigurable code, a large configuration is sent to the Montium at initialization and the reconfiguration overheads involve only small pieces of reconfigurable code. The small reconfiguration overhead enables Cognitive Radio to make quick adaption. However using static code, the reconfiguration requires reloading the complete configuration code. Therefore, from table 4.4 we conclude that the dynamic reconfiguration is about 10 times more efficient than loading a new configuration completely for every change.

Table 4.4: Bytes that need to be sent for reconfiguration

	reconfigurable code	static code
Initialization	1792	0
to 512-point	104	1138
to 256-point	82	1054
to 128-point	78	970
to 64-point	80	886
to 512-sparse (a nonzero <64 case)	78	–
Total	2214	4048

4.5 Chapter Summary

A novel sparse FFT, an efficient option for OFDM based Cognitive Radio where a large number of subcarriers are deactivated, is presented in this chapter. The proposed sparse FFT results in a higher computational efficiency compared to a radix-2 FFT when the nonzero ratio is less than half. The efficiency increases for a decreasing nonzero ratio. Compared to other similar algorithms, the proposed algorithm has the following advantages:

- it can deal with any nonzero distribution;
- it does not require conditional statements to determine the unnecessary computations;
- it is scalable with the number of nonzeros;
- it has a regular computational structure which facilitates a hardware implementation;
- it offers easy reconfigurability due to reuse of the radix-2 FFT.

As the core of an OFDM based Cognitive Radio, the reconfigurable sparse FFT has been mapped onto the proposed MPSoC platform. The design flow starts from the system-level description of the algorithm in TTL. The TTL design methodology offers several benefits for mapping Cognitive Radio onto the proposed MPSoC platform.

- The functional correctness of the proposed algorithm can be verified at system level.
- Profile information provided by TTL run-time environment is used for the complexity analysis at a system level.
- The TTL programming model facilitates the application partitioning and the task based implementation on MPSoCs.

A dynamically reconfigurable FFT module has been implemented on the Montium, a key element on the proposed platform. It enables two types of reconfiguration: 1) the reconfiguration of the FFT size; 2) the reconfiguration between sparse FFT and radix-2 FFT. The dynamically reconfigurable FFT is very efficient in terms of reconfiguration overhead which enables Cognitive Radio to make quick adaption. Although the sparse FFT implementation on the Montium is not as efficient as the theoretical predication due to the address generation overhead, it still gives a large saving in cases where the nonzero ratio is low.

Chapter 5

A Filter Bank Multicarrier Approach for Cognitive Radio

In spectrum pooling, OFDM is used as the baseband transmission scheme. A major challenge for OFDM based Cognitive Radio is its interference to the licensed system due to large sidelobes. Although a number of methods are proposed to mitigate the interference to the licensed system, they have disadvantages such as loss of bandwidth efficiency and high complexity. Therefore, other multicarrier techniques such as filter bank approaches are expected to be good alternative transmission schemes for Cognitive Radio. We will present an oversampled filter bank multicarrier scheme for Cognitive Radio in this chapter[1]*. The mapping of the proposed filter bank multicarrier system onto the Montium will be analyzed.*

[1] Parts of this chapter have been published in publications [QZ13].

5.1 Introduction

In spectrum pooling, OFDM is often used as the baseband transmission scheme. The cognition is realized by nullifying those subcarriers which cause interference to licensed users. The remaining frequency segments will be used optimally by Cognitive Radio. The additional benefit of OFDM is the reuse of the FFT module for spectrum sensing. However, due to the rectangular window in the time domain, the OFDM system has large sidelobes which cause interference to adjacent bands. This fact has also been recognized in [103]. They proposed two methods to mitigate the interference to the licensed user: deactivating more subcarriers adjacent to the licensed system or applying non-rectangular windows to reduce the spectrum leakage [104]. Both methods mitigate the interference at the cost of bandwidth efficiency. Moreover, the two methods did not consider the system implementation issues. Therefore, other multicarrier schemes could be interesting candidates for Cognitive Radio. This fact has also been observed in a recent publication [2]. A filter bank modulation called filtered multitone (FMT) [16] has been applied to very high-speed digital subscriber line technology to achieve high-level spectral containment in subchannels. This is also the key characteristic expected for a spectrum pooling system, where sideband power leakage should be kept to a minimum. In [119], we propose an oversampled filter bank multicarrier system for Cognitive Radio based on the idea of FMT.

The rest of the chapter is organized as follows. Section 5.2 gives a short introduction on filter bank multicarrier systems. The proposed oversampled filter bank multicarrier system for Cognitive Radio will be presented in section 5.3. A brief survey on other filter bank based approaches for Cognitive Radio will be given in section 5.4. The computational complexity of the filter bank multicarrier approach is much higher than the OFDM solution. Since the Montium on the proposed platform is targeted for such computationally complex algorithms, the mapping of the proposed filter bank multicarrier system onto the Montium is analyzed in section 5.6.

5.2 Filter Bank Multicarrier Basics

A filter bank is an array of bandpass filters that isolate different frequency components in a signal. The filter bank system often involves two processes: separation of the frequency components and recombination of the components to recover the original signal. The separation process is known as *analysis filter bank* and the recombination process is known as *synthesis filter bank*. Analysis and synthesis filter often appear in pairs and satisfy the perfect reconstruction condition which guarantees the perfect reconstruction of the signal. The filter bank system has many applications such as speech analysis and coding. For example, in speech coding the separate frequency components from the analysis filter bank are coded with different resolutions and recombined by the synthesis filter bank. In digital communications, there are also many applications of filter bank systems including equalization, high speed analog to digital conversion and transmultiplexing [96].

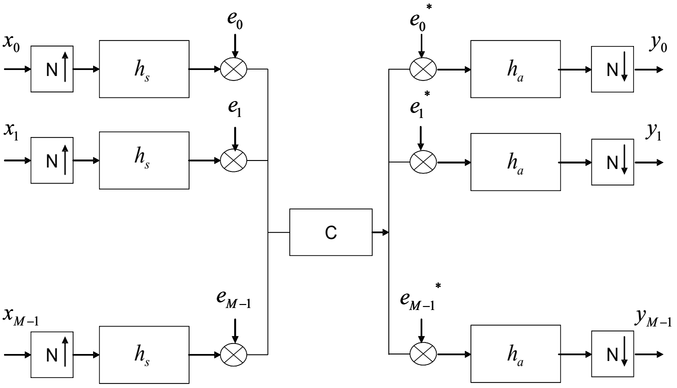

Figure 5.1: A multicarrier system based on filter banks

The basic idea of multicarrier transmission is to divide a broadband channel into parallel subchannels. The high-rate data stream is split into low-rate streams and transmitted on a subchannel. This transmission scheme can be modelled as a filter bank system structure (see figure 5.1) that resembles the transmultiplexer in [97]. At the transmitter, M complex symbols are first upsampled by a factor of N. The upsampling can be realized by inserting $N-1$ zeros between adjacent samples. Then the upsampled signal is filtered and combined by a synthesis filter bank. The synthesis filter bank can be obtained by one base band *prototype* filter h_s with proper frequency shifts. These frequency shifts can be seen as a modulation of M symbols on M subbands. The combined signal is transmitted to the channel. The receiver demodulates the signal by separating the combined signal into M subbands followed by the downsampling with a factor of N. In fact the receiver is an analysis filter bank obtained by proper frequency shifts from a prototype filter h_a. To perfectly reconstruct the signal, the filter bank has to satisfy the perfect reconstruction condition [96]. This implies that h_s and h_a are a pair of matched filters.

5.3 Oversampled Filter Bank Multicarrier for Cognitive Radio

In case of critical sampling ($N = M$) and the prototype filter is selected as a *sinc* shaped filter in the frequency domain, the multicarrier system becomes an OFDM system. When $N > M$, the filter bank system is called an *oversampled filter bank* (OSFB). The oversampling will increase intercarrier spacing by a factor of $\frac{N}{M}$ ($N > M$), see figure 5.2. In such a way intercarrier interference (see the overlapping part between two subcarriers) is largely reduced, which is the basic idea of FMT [16]. One would argue that the increased intercarrier spacing results in less subcarriers in a given bandwidth, thus losing bandwidth efficiency. However, compared with OFDM systems where extra cyclic prefix always has to be introduced, OSFB is no worse than OFDM in terms of bandwidth efficiency. Therefore, we think the OSFB is a good candidate for multicarrier based Cognitive Radio.

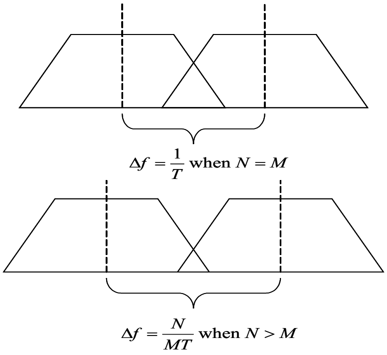
$\Delta f = \frac{1}{T}$ when $N = M$
$\Delta f = \frac{N}{MT}$ when $N > M$

Figure 5.2: Intercarrier spacing of a critically sampled and oversampled filter bank, where $\frac{1}{T}$ denotes the symbol rate

5.3.1 The System Model

The basic idea of multicarrier based Cognitive Radio is to deactivate the subcarriers causing interference to licensed users and optimally use the remaining part of the targeted spectrum. The deactivation can be realized by loading zeros on the intended subcarriers while others are loaded with modulated complex symbols at the transmitter which is an M band oversampled synthesis filter bank. An M band oversampled analysis filter bank on the receiver reconstructs the signal and sends only the symbols from active subcarriers for demodulation. The deactivation information is sent to both the transmitter and the receiver through a control channel. The simplified OSFB based multicarrier Cognitive Radio system is shown in figure 5.3. The modulation mode for the active subcarriers is adapted to the channel's SNR. The adaption can be done for all subcarriers as a whole or for each individual subcarrier based on subchannel's SNR. The adaptive bit loading for OFDM based Cognitive can be applied to the latter case.

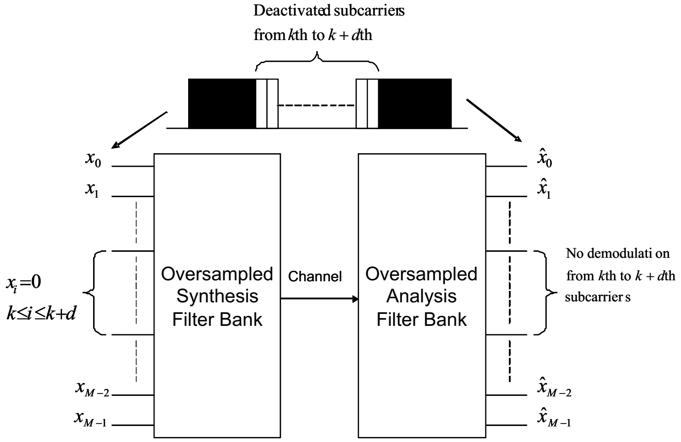

Figure 5.3: An OSFB multicarrier system for Cognitive Radio

5.3.2 Efficient Implementation Based on Generalized DFT Filter Bank

The implementation of the OSFB is not as straightforward as a critically sampled filter bank. The authors in [16] indicate to implement periodically time-varying filters in OSFB. However, this is difficult in practice. Therefore, we suggest an efficient implementation based on the generalized DFT filter bank (GDFT) model in [19].

The transmitted analog signal s_a can be expressed as a summation of M subband filter output:

$$s_a(t) = \sum_{m=0}^{M-1} \sum_{n=-\infty}^{+\infty} x_{m,a}(nT) h_{s,m,a}(t - nT) \tag{5.1}$$

where T denotes the symbol duration, $x_{m,a}(nT)$ is the symbol on mth subcarrier at nth instance, $h_{s,m,a}(t)$ is the analog synthesis prototype filter on the mth subband. Since the symbol rate is $\frac{1}{T}$, the sampling rate should be $\frac{N}{T}$. For each sampling instance k, the discrete signal $s(k)$ can be written as:

$$s(k) = \sum_{m=0}^{M-1}\sum_{n=0}^{+\infty} x_m(n)h_{s,m}(k-nN) = \sum_{n=0}^{+\infty} s_n(k-Nn) \tag{5.2}$$

where $x_m(n) = x_{m,a}(nT)$, $h_{s,m}(k)$ denotes the digitized synthesis filter and only symbol instances for $n \geq 0$ are considered. We define signal $s_n(k)$ at each instance n as:

$$s_n(k) = \sum_{m=0}^{M-1} x_m(n)h_{s,m}(k) \tag{5.3}$$

The subband filter $h_{s,m}(k)$ is derived from a real valued prototype filter $p(k)$ (e.g. raised cosine filter) by modulation as:

$$h_{s,m}(k) = p(k - \frac{L-1}{2})e^{j2\pi(m-\frac{M-1}{2})(k-\frac{L-1}{2})/M} \tag{5.4}$$

where L is the filter length, $\frac{M-1}{2}$ is set as carrier frequency and the delay of $\frac{L-1}{2}$ is introduced to make a causal system. Thus, $h_{s,m}(k) = 0$ for $k < 0$ or $k \geq L$. From eq. 5.3, we can see that $s_n(k)$ is the summation of the multiplication of M band symbols with L filter coefficients. Thus the length of $s_n(k)$ is L. We can write eq. 5.3 in a matrix form as:

$$s_n = H_s^T x_n \tag{5.5}$$

where x_n is the symbol vector and H_s is an $M \times L$ matrix:

$$H_s = [p(k - \frac{L-1}{2})e^{j2\pi(m-\frac{M-1}{2})(k-\frac{L-1}{2})/M}]_{M\times L} \tag{5.6}$$

A matrix multiplication of H_s^T and x_n can be done to produce signal s_n, however it costs $M \times L$ complex multiplications. To reduce the computational complexity, we reconstruct H_s from the $M \times M$ generalized DFT matrix T [19] and a diagonal matrix Λ_p where the diagonal holds L prototype filter coefficients as:

$$H_s = T \times [I_M \ (-1)^{M-1}I_M] \times [I_{2M} \ I_{2M} ... \ I_{2M} \ \hat{I}_{2M,u}] \times \Lambda_p \tag{5.7}$$

I_M and I_{2M} denote the $M \times M$ and $2M \times 2M$ identity matrices respectively and $\hat{I}_{2M,u}$ is the first u column submatrix of I_{2M}, where u is L modulo $2M$. The generalized DFT (GDFT) matrix T is expressed as:

$$T = \Lambda_1 W_M^* \Lambda_2 \tag{5.8}$$

where W_M denotes an M point DFT matrix. Λ_1 and Λ_2 are diagonal matrices where the ith diagonal elements for Λ_1 and Λ_2 are $e^{-j\pi(i-\frac{M-1}{2})(L-1)/M}$ and $e^{-j\pi i(M-1)/M}$ respectively. The same approach has also been introduced in [102]. Interested readers can also refer to Appendix A for a detailed mathematics derivation. From eq. 5.5, eq. 5.7 and eq. 5.8, we have:

$$\begin{aligned} s_n = \Lambda_p \times [I_{2M}\ I_{2M} ...\ I_{2M}\ \hat{I}_{2M,u}]^T \times \\ [I_M\ (-1)^{M-1} I_M]^T \times \Lambda_2 W_M^* \Lambda_1 x_n \end{aligned} \tag{5.9}$$

Similarly at the receiver, the recovered symbol $\hat{x}_n$ can be written in matrix form:

$$\hat{x}_n = H_a r_n \tag{5.10}$$

where $\hat{x}_n$ denotes the symbols from M bands and r_n is the received signal with length L. In order to satisfy the perfect reconstruction condition, the analysis filter matrix $H_a = H_s^*$ [96]. The recovered symbol $\hat{x}_n$ can be expressed as:

$$\begin{aligned} \hat{x}_n = \Lambda_1^* W_M \Lambda_2^* \times [I_M\ (-1)^{M-1} I_M] \\ [I_{2M}\ I_{2M} ...\ I_{2M}\ \hat{I}_{2M,u}] \times \Lambda_p r_n \end{aligned} \tag{5.11}$$

The generalized DFT implementation is based on eq. 5.9, eq. 5.2 and eq. 5.11, where the filter coefficient matrices consist of periodically varying GDFT matrices. Unlike the implementation in [16] where the coefficients are time varying, we can incorporate the periodicity by introducing two shift registers to store outputs and inputs. Thus, we can store the filter coefficients in fixed memory locations. This practice facilitates parallel implementations on hardware (e.g. the Montium). Figures 5.4 and 5.5 show the implementations of the GDFT filter bank transmitter and receiver respectively.

At the transmitter, M symbols are first transformed by T, which can be implemented as $2M$ phase shifts of complex numbers and an M point IFFT (here we consider that M is a power-of-two integer), based on eq. 5.8. The M transformed symbols $X_i (i = 0, 1.., M-1)$ are used to make sequence $\overline{X}_{2M} = [X_{i=0,1,...,M-1}\ -X_{i=0,1,...,M-1}]$. By repeating sequence $\overline{X}_{2M}$ $\lfloor L/M \rfloor$ ($\lfloor\rfloor$ denotes integer division) times and appending the first $L\ mod\ 2M$ elements in $\overline{X}_{2M}$ at the end, an L-element sequence is produced to be multiplied with L filter coefficients. The multiplication results are accumulated to a length L shift register D where each element equals zero at initialization. After the accumulation, the first N samples in D are shifted out as transmitted symbols and all other samples are shifted N positions ahead with N

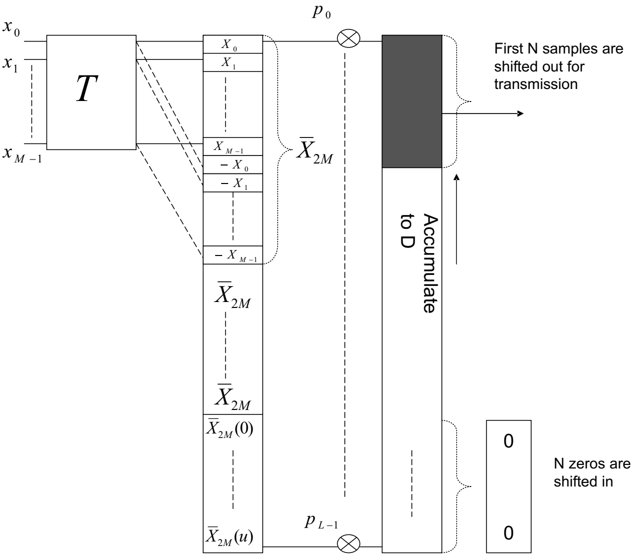

Figure 5.4: The GDFT filter bank transmitter implementation

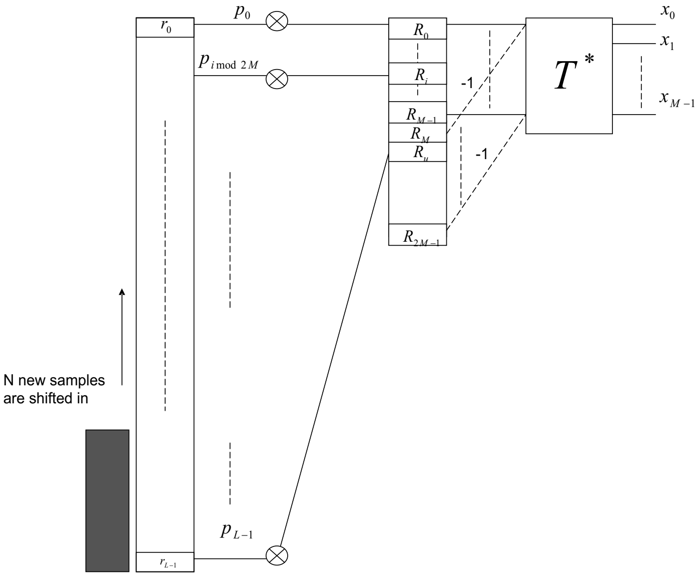

Figure 5.5: The GDFT filter bank receiver implementation

zeros shifted in. This accumulation and shifting operation is indicated in the last term of eq. 5.2 as $\sum_{n=0}^{+\infty} s_n(k - Nn)$.

At the receiver, L received symbols in a shift register are multiplied with L filter coefficients. The $i_{mod2M}(i = 0, 1, ..., L-1)$ multiplication results are combined to form a $2M$ sequence R. Then the second half of R is negated and combined with the fist half to produce M symbols to be transformed by T^* which is the conjugate of T. After the transform, M recovered symbols are obtained and N new symbols will be shifted in.

Based on the implementation, we made an analysis of the computational complexity by counting the number of complex multiplications. To transmit and receive each M symbols, we need $2L$ complex multiplications with filter coefficients, $4M$ for the phase shift, an M point IFFT and an M point FFT. The total computational complexity of the OSFB C_{OSFB} can be expressed as:

$$C_{OSFB} = 2L + 4M + 2 \times (\frac{M}{2} \log_2 M) \tag{5.12}$$

The computational complexity of the OFDM C_{OFDM} is:

$$C_{OFDM} = 2 \times (\frac{M}{2} \log_2 M) \tag{5.13}$$

From eq. 5.12 and eq. 5.13, we see that OSFB is computationally more complex than OFDM due to the extra filtering. Especially if the length of the prototype filter L is large, the computational complexity increases considerably. This complexity raises a question how to make efficient hardware implementations. Fortunately as shown in figure 5.4 and figure 5.5, the regular computational structures are good candidates for parallel processing.

5.3.3 Simulation Results

In the simulation, we consider a Cognitive Radio with 32 subcarriers where 8 are deactivated to avoid interference to licensed users. Both the OFDM based and the OSFB based multicarrier systems are considered. To simplify the evaluation of the system performance, the channel is assumed to be an Additive White Gaussian Noise (AWGN) channel with no distortion for both systems. The modulation scheme is considered to be the same for every subcarrier. In the OSFB, the oversampling ratio is chosen as $N = 36$. The square root raised cosine filter with rolloff factor 0.15 is assumed to be the prototype filter. The MATLAB function *rcosine* is used to generate the required prototype filter. We set the parameter group delay K as 9, thus the filter length $L = 2KN + 1 = 649$ (see help on MATLAB[2] function *rcosine*).

[2] Mathworks website: `http://www.mathworks.com`

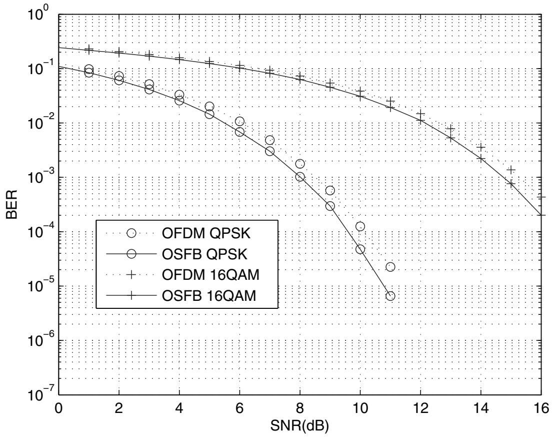

Figure 5.6: BER performance on AWGN

The group delay of a filter is a measure of the average delay of the filter and it is proportional to the filter length.

Figure 5.6 shows the BER performance comparison of OFDM and OSFB for the given scenario. For fair comparison, we assume no interference to both systems from licensed users. QPSK and 16QAM modulation schemes are considered. The BER performance of the OSFB is a little better than OFDM due to its less intercarrier interference. Figure 5.7 shows the transmitted spectrum for both systems. We can see the sideband power rejection of OSFB is much better than OFDM. The noise floor of the nullified spectrum is below -30dB for OSFB. It is much higher in case of OFDM therefore causing significant interference to licensed users. Even if more adjacent subcarriers are turned off in OFDM as indicated in [103], the interference to licensed bands is still high due to big sidelobes. Therefore, the traditional OFDM system without any modification is difficult to apply in a spectrum pooling system. The alternatives such as OSFB are more promising in the context of spectrum pooling.

5.3.4 The Design Tradeoffs

There are a number of tradeoffs in designing an OSFB multicarrier system such as choices of different prototype filters, the length of filters and oversampling ratio. These tradeoffs have impact on the BER performance, sideband power rejection, bandwidth efficiency and computational complexity. Here we discuss some tradeoffs based on our system in the simulation.

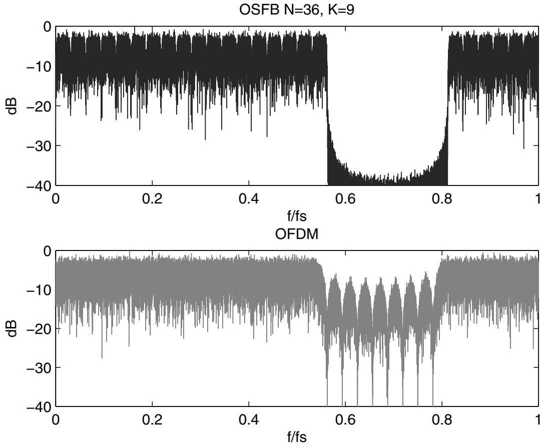

Figure 5.7: Transmitted spectrum with null subcarriers

- **The choice of group delay K**

  In figure 5.8 and figure 5.9, we show the BER performance and the transmitted spectrum of the OSFB by increasing the group delay K of the prototype filter from 4 to 9 while keeping the rolloff factor and the oversampling ratio unchanged. QPSK modulation is used and an AWGN channel is assumed. With the same sampling ratio, the group delay K is a good indication of the prototype filter length. Increasing K from 4 to 9 will increase the prototype filter length L from 289 to 649. For an increasing K, we can see an improvement in the BER performance (figure 5.8) and especially the sideband rejection (figure 5.9). This is largely due to the high order prototype filter giving better frequency response and sideband rejection. However, the computational complexity increases by 90% according to eq. 5.12. Therefore, the length of the prototype filter is an important tradeoff between the system performance and the computational complexity.

- **The choice of oversampling ratio N**

  Intuitively, increasing the oversampling ratio will result in larger intercarrier spacing. The increased intercarrier spacing tends to reduce the power leakage to adjacent subcarriers, therefore better BER performance and less interference to licensed bands are expected. If we consider only one side of the spectrum leakage power P_{leak} generated

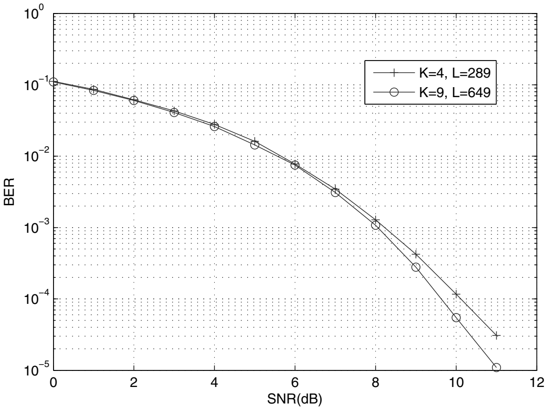

Figure 5.8: BER performance of the OSFB with different K and QPSK modulation in AWGN

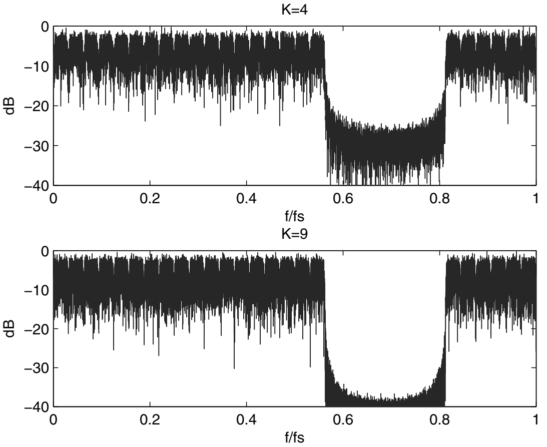

Figure 5.9: Transmitted spectrum of the OSFB with different K

by one subcarrier, it can be defined as:

$$P_{leak} = \int_{\frac{\Delta f}{2}}^{+\infty} \Phi(f)df \tag{5.14}$$

where $\Phi(f)$ denotes the power density spectrum of a subcarrier and Δf is the subcarrier spacing. If an ideal raised cosine filter is used as prototype filter in OSFB and the transmit power of symbols on each subcarrier is normalized to 1, the power density spectrum of a subcarrier $\Phi(f) = |H_{rc}(f)|^2$. Thus, eq. 5.14 can be expressed as:

$$P_{leak} = \frac{1}{T} \int_{\frac{N}{2MT}}^{\frac{1}{T}} |H_{rc}(f)|^2 df \tag{5.15}$$

where T is the symbol duration and $H_{rc}(f)$ denotes frequency response of a raised cosine filter. Ideally from eq. 5.15, the power leakage is reduced to zero when the sampling ratio $N > 2M$ regardless of the choice of rolloff factor. However, in practice the ideal raised cosine filter can never be realized therefore the spectrum power leakage will always exist. Nevertheless eq. 5.15 can still serve as a rough guidance to make tradeoffs between the oversampling ratio and the sideband power leakage in the OSFB based on the raised cosine prototype filter. However, increasing the oversampling ratio obviously leads to bandwidth inefficiency. Furthermore, the increased oversampling ratio will also result in an increase of computational complexity.

- **The type of prototype filter**

  In our discussion, only the raised cosine filter is considered. However, there is freedom for prototype filter design if the reconstruction error is on an acceptable level. Much research has been done on prototype filter design for filter bank systems. For example in [52], a small sidelobe prototype filter may be an interesting option for a filter bank multicarrier system for Cognitive Radio.

5.4 Other Filter Bank Based Multicarrier Systems

In the context of Cognitive Radio, other filter bank approaches based multicarrier can also be considered such as Cosine Modulated Filter Bank (CMFB) [2] and Wavelet Packet based Multicarrier Modulation (WP-MCM) [48].

Cosine Modulated Filter Bank

A CMFB for wireless communications is considered in [25]. A CMFB does not resort to oversampling to reduce the intercarrier interference. Instead, it applies critical sampling but resorts to a well designed filter that can perfectly separate the overlapped adjacent bands. The critical sampling results in higher bandwidth efficiency because no extra samples have to be introduced. The other advantage of CMFB claimed in [25] is its blind equalization capability. Blind equalization is allowed by using real valued symbols (Pulse Amplitude Modulated (PAM) symbols) for each subcarrier. However, there are some drawbacks of CMFB. Firstly, transmitting PAM symbols on subcarriers results in throughput loss compared to transmitting QAM symbols. Secondly, the prototype filter has to be carefully designed. Lastly, the implementation of CMFB is not as straight forward as OSFB.

Wavelet Packet based Multicarrier Modulation

Wavelet transform and filter banks are closely related. Filter banks may have nonuniform bandwidth for each subband and nonuniform decimation ratios in each subband. These two nonuniformities can be considered as the fundamental ingredients in wavelet transform [96]. In [48], a novel wavelet based multicarrier system was proposed for spectrum pooling. The proposed wavelet multicarrier system is based on multistage tree-structured paraunitary filter banks. They applied the idea of frequency selective filter banks to design wavelet packet bases which guarantee strict spectral containments. Furthermore, these wavelet packet bases have to be orthonormal to ensure perfect reconstruction. The simulation results in [48] show that a well designed wavelet multicarrier system can significantly reduce the interference of Cognitive Radio to the licensed user as well as the interference of the licensed user to Cognitive Radio. However, the wavelet packet bases have to be generated every time when Cognitive Radio adapts to its new environment such as the update of spectrum occupancy information and changing channel conditions. This leads to more system complexity on top of the filter bank implementation.

5.5 Challenges

Although a filter bank multicarrier system is a promising option in the context of Cognitive Radio, there are still a number of challenges for the practical system.

Channel Equalization

Filter bank multicarrier systems have been applied in Digital Subscriber Line (DSL) standards. An example is the FMT transceiver in [16] which has been included in the Very-high-bit-rate Digital Subscriber Line (VDSL) standard [98]. However, filter bank multicarrier is not a common practice in wireless communications. A major challenge for filter bank multicarrier based wireless systems is the channel equalization since adequate channel equalization techniques are not yet available for filter bank based multicarrier systems [42]. In FMT [16], they propose a separate decision feedback equalizer for each subcarrier. Such a decision feedback equalizer is very computationally expensive which makes it difficult for use in wireless communications. Although a blind equalization method was proposed in a CMFB [25] system for wireless applications, it is based on the assumption that each subchannel can be approximated as a flat fading channel. Therefore, channel equalization is still an open issue for filter bank multicarrier systems. When considering filter bank multicarrier for Cognitive Radio, channel equalization is even more complex because Cognitive Radio requires more adaptivity to changing channel conditions.

Timing Sensitivity

In [71], the performance comparison of a filter bank multicarrier system and an OFDM system under fading channel shows that the filter bank multicarrier system is more sensitive to timing delay than the OFDM system. The reason is that a fading channel with a large delay spread will cause time mismatch of input samples with filter coefficients in filter bank multicarrier systems. In [62], Nikookar et. al. investigated the effects of timing offsets on the wavelet based multicarrier system. The performance comparisons with classical OFDM indicate that the wavelet based multicarrier system is very sensitive to even small timing discrepancies between transmitter and receiver. Thus, sensitivity to timing offset is recognized as a major drawback of the wavelet based multicarrier system. Moreover, most synchronization algorithms devised for multicarrier systems are mainly targeting the popular OFDM format and cannot be directly applied to filter bank multicarrier systems. Therefore, how to deal with the timing offset in filter bank multicarrier systems is another open challenge.

Complexity

The complexity analysis in [119] and [71] show that the implementation costs of filter bank multicarrier systems are generally higher than FFT based OFDM systems. The additional costs stem mainly from the filtering process. If channel equalization is considered, the complexity will become even higher [50]. Therefore, complexity is another major drawback of filter bank based multicarrier systems.

5.6 Mapping onto the Reconfigurable Platform

The computational complexity of the filter bank multicarrier approach is much higher than the OFDM solution. Since the Montium is targeted for such computationally complex algorithms, we will analyze how the proposed scheme can be mapped onto the Montium based platform. The mapping is based on the GDFT based implementation model in section 5.3.

5.6.1 The GDFT Filter Bank on the Montium

From figure 5.4 and figure 5.5, we see that both the synthesis filter bank at the transmitter and the analysis filter bank at the receiver consist of two major computation tasks: the generalized DFT transform and the multiply-accumulate (MAC) operations with the filter coefficients.

GDFT transform

The difference between the generalized DFT and the normal DFT is that the GDFT introduces phase shifts. According to eq. 5.8, the phase shifts can be implemented as multiplications with two complex vectors. The implementation of the GDFT matrix T for the synthesis filter bank on the transmitter side is shown in figure 5.10. Similarly, the inverse GDFT matrix T^* for the analysis filter bank on the receiver side can be implemented as shown in figure 5.11.

The GDFT transform can be mapped onto one Montium processor by combining the phase shifts with an existing FFT/IFFT module. As indicated in [39], the FFT/IFFT module on the Montium uses ALU1 through ALU4 for computations. All 10 memories are used during the computation. Two memories, MEM9 and MEM10, are used to store the real and the imaginary part of the twiddle factors, respectively. The remaining memories, MEM1 to MEM8, are used to store the (intermediate) results. In order

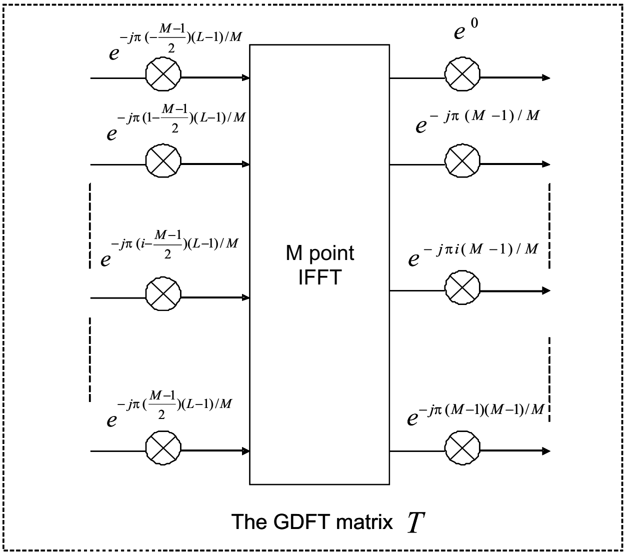
$e^{-j\pi(-\frac{M-1}{2})(L-1)/M}$
e^{0}
$e^{-j\pi(1-\frac{M-1}{2})(L-1)/M}$
$e^{-j\pi(M-1)/M}$
$e^{-j\pi(i-\frac{M-1}{2})(L-1)/M}$
M point IFFT
$e^{-j\pi i(M-1)/M}$
$e^{-j\pi(\frac{M-1}{2})(L-1)/M}$
$e^{-j\pi(M-1)(M-1)/M}$
The GDFT matrix T

Figure 5.10: The implementation of the GDFT matrix T on the transmitter side

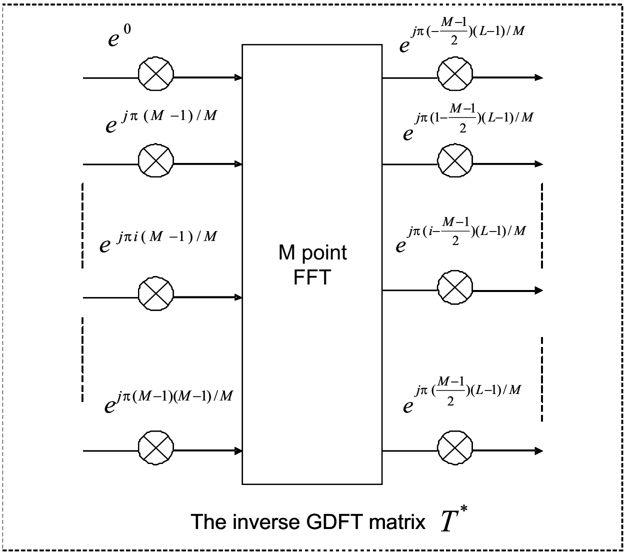
e^{0}
$e^{j\pi(-\frac{M-1}{2})(L-1)/M}$
$e^{j\pi(M-1)/M}$
$e^{j\pi(1-\frac{M-1}{2})(L-1)/M}$
$e^{j\pi i(M-1)/M}$
M point FFT
$e^{j\pi(i-\frac{M-1}{2})(L-1)/M}$
$e^{j\pi(M-1)(M-1)/M}$
$e^{j\pi(\frac{M-1}{2})(L-1)/M}$
The inverse GDFT matrix T^{*}

Figure 5.11: The implementation of the inverse GDFT matrix T^{*} on the receiver side

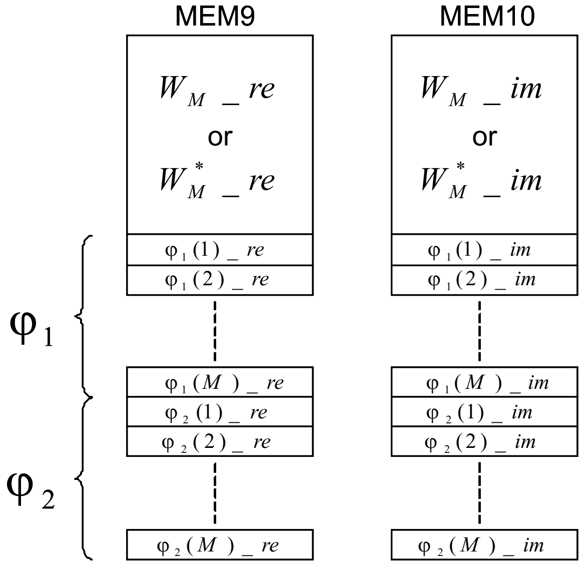

Figure 5.12: MEM9 and MEM10 for the GDFT transform

to use the existing FFT/IFFT module on the Montium without modification, the phase shift vectors φ_1 and φ_2 can be stored in MEM9 and MEM10 appended to the twiddle factors of the FFT/IFFT as shown in figure 5.12. $\varphi_1 = e^{-j\pi(i-\frac{M-1}{2})(L-1)/M}$ and $\varphi_2 = e^{-j\pi i(M-1)/M}$ $(i = 0, 1, ..., M-1)$ for the GDFT implementation. $\varphi_1 = e^{-j\pi i(M-1)/M}$ and $\varphi_2 = e^{-j\pi(i-\frac{M-1}{2})(L-1)/M}$ $(i = 0, 1, ..., M-1)$ for the inverse GDFT implementation. Both phase shift vectors are complex and have M elements. For the multiplication of the input samples with the first phase shift vector φ_1, the memory addressing of MEM9 and MEM10 starts from the first element of φ_1 and increases by 1 until the last element of φ_1. After the multiplication with φ_1, the memory address is set to the first memory position for the FFT/IFFT computation such that the existing FFT/IFFT module on the Montium can be used without modification. When the FFT/IFFT is done, the memory address is set to the position of the first element of φ_2 for the multiplication with φ_2.

A phase shift is just a simple complex multiplication which can be done by 4 Montium ALUs (eg. ALU1 through ALU4) in one clock cycle. Figure 5.13 shows a detailed picture of a phase shift implementation on the Montium. A possible memory allocation for input, output and phase shift vector is indicated in figure 5.13. The operations on ALUs shows how one complex multiplication is produced by combining 4 real multiplications. One

ALU can add or subtract the multiplication result from adjacent ALU via west-to-east link. The intermediate results after the multiplication with the first phase shift vector should be stored in place for the FFT/IFFT module so that no clock cycle is wasted for moving data. Therefore, the phase shifts cost $2M$ clock cycles. An M point FFT/IFFT module, with M a power-of-two, costs $(\frac{M}{2}+2)\log_2 M$ clock cycles on the Montium [39]. The computation of a GDFT costs $2M+(\frac{M}{2}+2)\log_2 M$ clock cycles in total on the Montium.

The number of subcarriers of a filter bank system can be a reconfigurable parameter to provide different throughput and to adapt to different channel conditions. Therefore, the Montium should support different sized GDFTs. Since the Montium can support variable sized radix-2 FFT, different sized GDFTs can be easily implemented by only introducing additional phase shifts. In our mapping scheme, the maximum supported power-of-two GDFT size on one Montium is 256, which is constrained by the sum of the FFT/IFFT twiddle factors and the phase shift vectors in MEM9 and MEM10. If we consider a 512 point GDFT, only two phase shift vectors will require the entire 1024 memory space in the Montium thus it leaves no space to store twiddle factors. Note that when two Montiums are used; one for the FFT/IFFT and the other for the phase shift, the maximum supported power-of-two GDFT size is 1024.

Filter MAC

The GDFT computation requires a relatively large configuration space and memory on the Montium. One Montium processor may not offer enough configuration space and memory for the complete filter bank system when M is large. Therefore, the multiply-accumulate (MAC) operations with the filter coefficients can be performed by another Montium. The NoC on the platform can transport the intermediate data between two Montiums.

Since the Montium offers plenty of computation and storage resources (5 ALUs and 10 memories), we have relatively a large freedom to implement the MAC operations with the filter coefficients. Therefore, there are different mapping schemes. Here, we show an example of mapping the filter MAC at the transmitter. The computational model used for mapping is based on figure 5.4.

- The real and imaginary parts of M complex samples from the GDFT are stored in MEM1 and MEM5 respectively. The prototype filter coefficients p_i are all real values and they can be stored in one memory

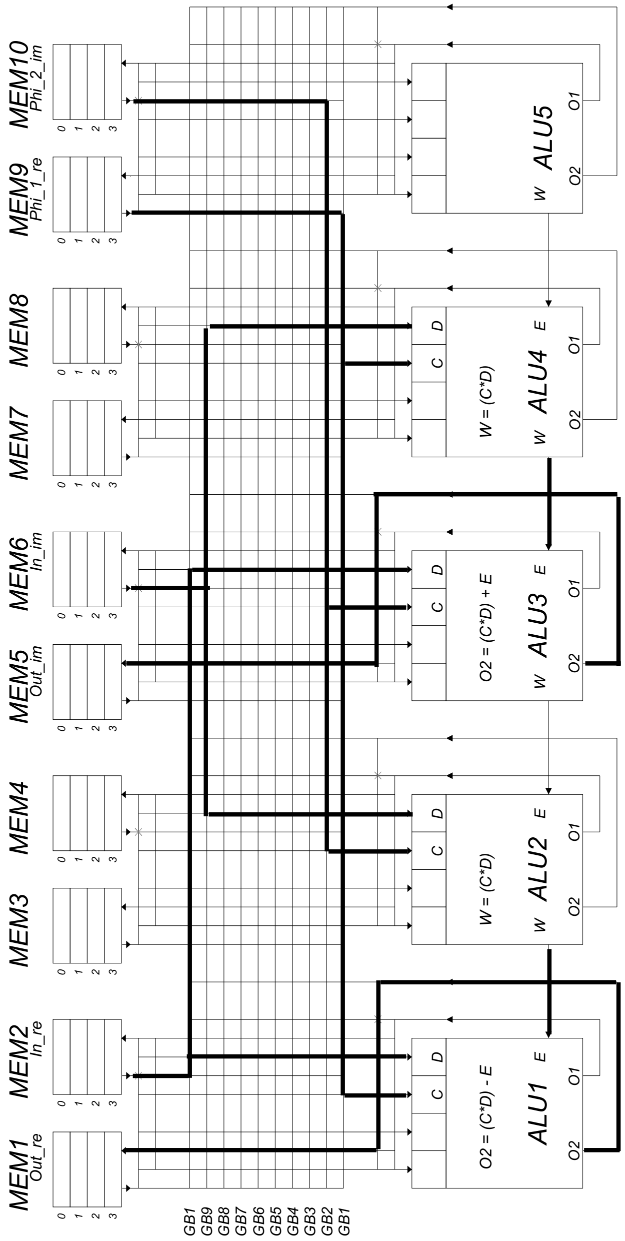

Figure 5.13: An phase shift implementation of the GDFT on the Montium

(e.g. MEM9). Since p_i is predetermined, the negation of the GDFT output X_i can be incorporated into the prototype filter coefficients such that $P = [p_1, p_2 ... p_M, -p_{M+1}, -p_{M+2}, ..., -p_{2M} ... (-1)^{\lfloor L/M \rfloor} p_L]$. MEM3 and MEM7 can be used as the shift register D in figure 5.4 for real and imaginary values respectively.

- ALU1 does the following MAC operation: multiply the real part of X_i (in MEM1) with the filter coefficients (in MEM9) and add the real value from D (in MEM3). ALU3 does the same MAC operation for the imaginary values. Thus, one filter MAC operation takes one clock cycle. Since L such MAC operations are required, it will takes L clock cycles to complete the filter MAC on the Montium. The real and imaginary results of the MAC operation are stored in MEM4 and MEM8 respectively. The first N values in MEM4 and MEM8 can be read out for transmission. Then MEM4 and MEM8 will serve as the shift register D and the initial address starts from position $N + 1$ to introduce the shift.

- The next M samples from the GDFT will be stored in the same memory location. The MAC operation will be the same, but the only difference is that the accumulated values are from MEM4 and MEM8. The results will be stored in MEM3 and MEM7. Therefore, MEM3 and MEM4 alternately serve as the shift register D for real values. So do MEM7 and MEM8 for the imaginary part of the values.

As discussed in section 5.3, the order of the prototype filter can be varied to offer tradeoffs between performance and complexity. Moreover, different prototype filters can be implemented to achieve optimum sideband rejection and BER performance. Therefore, the prototype filter coefficients should be reconfigurable. This reconfiguration can be easily supported by the Montium just by reloading the coefficients in the memory.

As an experiment, an oversampled filter bank multicarrier transmitter is mapped onto two Montiums. The parameters are taken from the simulation model in section 5.3, where the number of subcarriers $M = 32$, the oversampling ratio $N = 36$ and the length of the prototype filter $L = 649$. We simulated the GDFT task and the filter MAC task on a single Montium tile with the Montium simulator. Table 5.1 gives an overview of the number of processor cycles required to calculate one data symbol (32 samples) for each task. The result in Table 5.1 indicates that the filter MAC is more computationally intensive than the GDFT transform. If the Montium run at 100MHz, the processing time for the GDFT task and the filter MAC task

Table 5.1: The number of cycles for the GDFT synthesis filter bank on the Montium

Task	#cycles
GDFT	156
Filter MAC	651

takes about 1.56μs and 6.51μs respectively in this case. We may conclude that mapping the filter bank multicarrier onto the Montium results in acceptable latency. Moreover, the Montium supports a reconfigurable filter bank system. Therefore, the Montium is a good candidate architecture to support an adaptive filter bank multicarrier system for Cognitive Radio.

5.7 Chapter Summary

Due to large sidelobes, OFDM based Cognitive Radio has large spectrum leakage which may cause potential interference to licensed users. Therefore, a filter bank approach is considered as an interesting alternative to OFDM in the context of Cognitive Radio due to its good spectrum containment. As one of contributions of this thesis, an oversampled filter bank multicarrier system Cognitive Radio is presented in this chapter as a solution to mitigate the interference to the licensed system.

The reason to apply the oversampled filter bank is that the oversampling increases intercarrier spacing which reduces the overlapping of subcarriers. As a result, the spectrum leakage of one subcarrier to adjacent subcarriers can be reduced. A complete oversampled filter bank multicarrier system for Cognitive Radio is proposed. The proposed OSFB Cognitive Radio system is implemented based on the generalized DFT model. We compared the performance of the OSFB based Cognitive Radio where 8 out of 32 subcarriers are deactivated with its OFDM counterpart. The BER performance of the OSFB based Cognitive Radio is sightly better than the OFDM based Cognitive Radio, but the sideband rejection improves significantly.

However, we recognized a number of challenges for filter bank multicarrier systems. A major challenge for filter bank multicarrier based wireless systems is channel equalization. Moreover, the filter bank multicarrier system is more sensitive to timing delay than the OFDM system. The other disadvantage of a filter bank multicarrier system is that its complexity is generally higher than the FFT based OFDM system due to the additional filtering.

Due to the high computational complexity of a filter bank system, a general purpose solution can not meet the requirements of processing speed and energy consumption. An ASIC solution may meet these requirements, however it can not support an adaptive system. Therefore, we consider a coarse-grain reconfigurable solution, such as the Montium, as a good candidate to support such computationally intensive algorithm with reconfigurable options. We analyzed the mapping of the proposed filter bank system onto the Montium. Based on the GDFT implementation, the filter bank system consists of two major computational tasks: the GDFT transform and the filter MAC. We propose to use two Montiums: one for the GDFT and the other for the filter MAC. The GDFT is a combination of a FFT/IFFT and the multiplication with two phase shift vectors. It can be easily adapted from an existing FFT/IFFT implementation on the Montium. As an experiment, an OSFB multicarrier transmitter is implemented on the Montium. The simulation results show that the filter MAC task is computationally more intensive than the GDFT task. The cycle count on the Montium indicates that the Montium can process the GDFT and filter MAC with a acceptable latency. Moreover, the Montium can support different subcarrier sizes and various prototype filters.

Chapter 6

Spectrum Sensing

The transmission of Cognitive Radio strictly depends on the reliable detection of the primary system through spectrum sensing. As a result, spectrum sensing is an essential part of Cognitive Radio. The physical layer spectrum sensing is a fundamental step to establish the spectrum occupancy information. The major task of the physical layer spectrum sensing is to detect the licensed signal by employing various signal processing techniques. Two categories of signal processing techniques are particularly interesting for the spectrum sensing scheme of Cognitive Radio, namely energy detection and feature detection. This chapter[1] provides an insight into these signal processing techniques for spectrum sensing.

[1] Parts of this chapter have been published in publication [QZ12].

6.1 Introduction

As the secondary user of unoccupied spectrum, Cognitive Radio does not have the a priori right to use the spectrum. Its communication should be strictly conditional on the reliable detection of the primary user transmissions. As a result, Cognitive Radio has to perform frequent measurements on primary user's activity through spectrum sensing, which is considered as an essential part of Cognitive Radio. From a system perspective, spectrum sensing is a cross-layer design issue. First of all, it requires a Radio Frequency (RF) part which can work in a wide range of frequency bands and receive adequate signal strengths from these bands. Secondly, on the physical layer level the received signal has to be processed to detect the primary user's signal. Finally, a mechanism on higher layers has to be implemented to combine the sensing results from individual Cognitive Radio nodes and make a joint decision on the spectrum allocation.

Since the focus of this thesis is on the physical layer of Cognitive Radio, this chapter treats the topic of spectrum sensing mainly from a physical layer signal processing perspective. There are two requirements for physical layer sensing: sensing reliability and sensing time. Reliable sensing means the successful detection of the active primary system. Unreliable sensing on the other hand misses the detection of the primary user's activity thus results in harmful interference. The other kind of unwanted detection is false alarm which compromises the efficiency of Cognitive Radio. Sensing time should be minimized so that Cognitive Radio can quickly identify the licensed user and adapt its transmission accordingly. Moreover, short sensing time may result in less computational burden on the radio platform. However, these two requirements are often contradictory. To obtain a better sensing result, the sensing unit may need to process more samples or apply more complex DSP algorithms. As a result, better sensing reliability is at the expense of longer sensing time and/or more computations. On the other hand, short sensing time and/or simple algorithms may result in less reliable detection. Therefore, the major challenge for physical layer sensing is to design a sensing scheme which can reliably detect the primary user's activity with a minimum sensing time.

From the signal processing perspective, spectrum sensing can be treated as a signal estimation and detection problem. The traditional techniques in signal detection and estimation can be applied to identify the primary user's signal in the context of Cognitive Radio. Cabric et. al. mentioned three types of detection techniques for spectrum sensing in [10], namely matched filtering, energy detection and feature detection.

6.1 Introduction

Matched filtering is an optimal way for signal detection in communication systems. It correlates the received signal with a known signal pattern which maximizes the received signal to noise ratio. However, it requires a priori knowledge of the licensed signal. Moreover, strict timing and synchronization has to be guaranteed to coherently detect the licensed signal. In the case where the licensed system is determined, matched filtering is applicable. For example, in the IEEE 802.22 standard, spectrum sensing can take advantage of the pilot information of the DTV signal which is known to Cognitive Radio. In [14], a matched filtering method is used to detect the DTV signal by correlating the received signal with the Field Sync segment, a pilot information sequence, of the DTV signal. However, the authors in [14] also recognized that the detection is severely affected by synchronization errors and channel impairments. Moreover, the complexity of the proposed method is extremely high due to the fact that the correlation involves a large number of samples. If Cognitive Radio is considered in a more general context, adequate knowledge of the licensed signal is often not available to Cognitive Radio and there could be different licensed systems on the targeted spectrum. Therefore, either it is impossible to perform matched filtering or it is too complex to implement a detector to match all possible licensed signals. Since no specific licensed system is considered in the context of the AAF project, matched filtering is not an option thus out of the scope of this thesis.

Energy detection has been the most common detection scheme due to its simplicity in both concept and practice. An energy detector, also known as radiometer, is the optimal detector when only the power of the signal and noise is known to the receiver [95]. Therefore, energy based detection is the heart of our study on spectrum sensing. In section 6.2, energy detection will be discussed in detail. Implementing a reliable yet efficient energy detector is the theme of our discussion. Under this theme, the contributions of this thesis to spectrum sensing are presented.

Due to signal fading or raised noise levels, the licensed signal could be completely drowned in the noise. In such low SNR scenarios, energy detection is inadequate to reliably detect the licensed signal. It is beneficial to exploit the feature of the licensed signal to enhance the reliability of the detection. In such a context, cyclostationary feature detection is proposed to detect weak licensed signals [10]. We will introduce feature detection in section 6.3 and present its advantages and disadvantages. Mapping the spectrum sensing algorithms on the proposed platform is considered in section 6.4 Finally, we will summarize the chapter and make some recommendations on physical layer spectrum sensing in section 6.5.

6.2 Energy Detection

Energy detection is the most commonly used method to detect signals in presence of noise. Two advantages make it the primary choice for the spectrum sensing scheme of Cognitive Radio. First, it requires minimum knowledge of the signal. Second, a simple digital implementation can be realized through an FFT algorithm that enables fast sensing and easy integration with an OFDM based transmission system.

6.2.1 Theoretical Background

Theoretically, energy detection is formulated as a hypothesis testing problem. The detection of the licensed signal is the test of the following two hypotheses: under hypothesis H_0 the licensed signal is not present and only the noise is received, and under hypothesis H_1 the licensed signal is present in the noise. Assume the licensed signal has been successfully received, downconverted and sampled. The above hypotheses can be expressed in the discrete form as:

$$H_0 : y[k] = n[k] \quad k = 1, ..., N \tag{6.1}$$

$$H_1 : y[k] = s[k] + n[k] \quad k = 1, ..., N \tag{6.2}$$

where the signal $s[k]$ and the noise $n[k]$ are Gaussian distributed independent random variables with zero means and variance σ_s^2 and σ_n^2 respectively. A decision statistic is the energy measured over N samples:

$$\epsilon(y) = \sum_{k=1}^{N} y[k]^2 \tag{6.3}$$

Because the decision statistic is a sum of N squared gaussian random variables, its Probability Distribution Function (PDF) is chi-square χ_N^2 [95]. The measured energy $\epsilon(y)$ is compared with a threshold γ to determine the presence of the signal. If $\epsilon(y) > \gamma$ under the hypothesis H_1, we assume the signal is detected. If $\epsilon(y) > \gamma$ under hypothesis H_0, it means the detection gives a false alarm for the signal which is actually not present. In detection theory, the probability of detection P_d and the probability of false alarm P_{fa} are two important performance metrics for the detector. They are defined as:

$$P_d = P(\epsilon(y) > \gamma | H_1) \tag{6.4}$$

$$P_{fa} = P(\epsilon(y) > \gamma | H_0) \tag{6.5}$$

A threshold γ can be determined to meet the required P_d or P_{fa}. When the number of the samples N is large, the chi-square distributed decision statistic can be approximated with a Gaussian distribution whose mean and variance are N and $2N$ respectively. Then, P_d and P_{fa} can be expressed as:

$$P_d = Q(\frac{\frac{\gamma}{\sigma_s^2+\sigma_n^2} - N}{\sqrt{2N}}) \tag{6.6}$$

$$P_{fa} = Q(\frac{\frac{\gamma}{\sigma_n^2} - N}{\sqrt{2N}}) \tag{6.7}$$

, where $Q()$ denotes the Q function defined in eq. 3.5. If the detector has to achieve a specific P_d and P_{fa} and the signal to noise ratio SNR is defined as $\frac{\sigma_s^2}{\sigma_n^2}$, the relation between the number of samples N and the SNR can be derived from eq. 6.6 and eq. 6.7 by cancelling γ as:

$$N = 2[\frac{Q^{-1}(P_{fa}) - Q^{-1}(P_d)}{SNR} - Q^{-1}(P_d)]^2 \tag{6.8}$$

From eq. 6.8, we can see the number of samples N that needs to meet a certain performance requirement is inversely proportional to the SNR^2. Explained in a simple way, detecting the licensed signal received with better SNRs requires less samples. If the received licensed signal has a very bad SNR, the number of samples needed has to be considerably large, which makes energy detection virtually impossible. This fact implies that energy detection may be inadequate in bad SNR scenarios. Furthermore, in [92], Tandra et al. argue that when the noise level is uncertain, the energy detection will reach the so called SNR wall beyond which the detection is impossible no matter how many samples are considered.

In spectrum sensing, we are more interested in the signal energy distribution in the frequency domain, which is characterized by the power spectral density. If a discrete signal is considered, the power spectral density $P(w)$ can be defined as:

$$P(w) = \lim_{N\to\infty} E\{\frac{1}{N}|\sum_{n=0}^{N-1} x(n)e^{-jwn}|^2\} \quad w \in [-\pi \ \pi] \tag{6.9}$$

One of the basic methods to estimate the power spectral density with a finite number of samples is to calculate a periodogram [36]. The estimated power spectral density $\hat{P}_p(w)$ represented by the periodogram is defined as:

$$\hat{P}_p(w) = \frac{1}{N}|\sum_{n=0}^{N-1} x(n)e^{-jwn}|^2 \quad w \in [-\pi \ \pi] \tag{6.10}$$

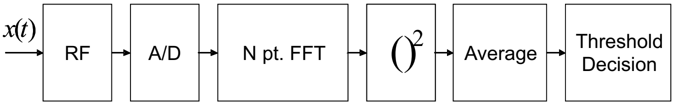

Figure 6.1: FFT based energy detection scheme

In practice, $\hat{P}_p(w)$ can be computed via an FFT. The power spectrum estimation will serve as the decision statistic for the threshold detection. This FFT based energy detection scheme is described in figure 6.1. The digitized samples are processed with an FFT, squared and averaged to obtain the estimated power spectral density according to eq. 6.10. This estimation will be compared with a threshold to determined the presence of the signal in the background of the noise. In this scheme, we assume the front-end is a wide-band receiver which targets the band of interest for Cognitive Radio. From a computational point of view, the FFT processing is the core of the scheme. The size of the FFT is a critical parameter because a larger FFT size improves the bandwidth resolution and the processing gain (same P_d and P_{fa} at lower SNR) but increases the sensing time.

6.2.2 Multi-resolution Spectrum Sensing

In the FFT based energy detection, it is common to choose a fixed FFT size to meet the desired resolution. The merits of multi-resolution spectrum sensing for Cognitive Radio have not been fully explored until the treatment in the recent literature [41, 61]. Although different methods have been applied in these papers, the basic idea is the same. The total bandwidth is first sensed using a coarse resolution. Fine resolution sensing is performed on a portion of the bands of interest for Cognitive Radio. In such a way, Cognitive Radio avoids sensing the whole band at the maximum frequency resolution. Therefore, the sensing time is reduced and power has been saved by avoiding unnecessary computations. Hur et al. [41] proposed a wavelet based multi-resolution sensing technique in the analog domain. Neihart et al. [61] discuss an FFT based multi-resolution spectrum sensing for multiple antenna Cognitive Radio. In [61], the multiple antenna architecture enables parallel processing at different frequencies and thus reduces the sensing time. However, it increases the chip area and power consumption. Moreover, the mixer has to generate multiple frequencies for coarse resolution sensing and has to switch to a single frequency while starting fine resolution sensing. Inspired by their work, we proposed a multi-resolution sensing scheme for a single antenna Cognitive Radio receiver in [118].

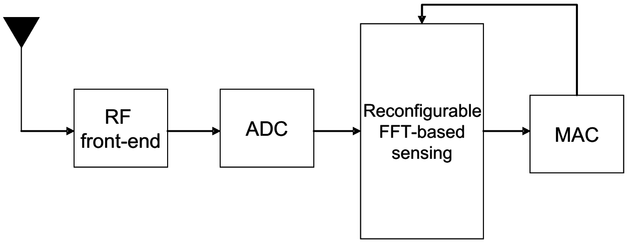

Figure 6.2: Block diagram of reconfigurable FFT based multi-resolution sensing

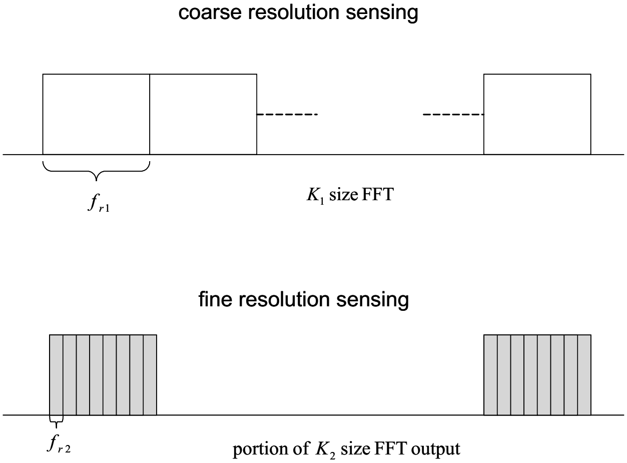

Figure 6.3: A multi-resolution sensing example

Proposed Scheme

In the proposed scheme, we assume a single antenna receiver for spectrum sensing as shown in figure 6.2. Furthermore, we assume the receiver can digitize the total targeted bandwidth of B_{tot} which is unlike the assumption in [61] where the total bandwidth is split into subbands which are digitized separately. If K is the size of the FFT which produces the spectrum, the frequency resolution $f_r = \frac{B_{tot}}{K}$. Figure 6.3 shows an example of multi-resolution sensing. A coarse resolution sensing is done by using a smaller K_1 size FFT with a resolution $f_{r1} = \frac{B_{tot}}{K_1}$. The energy on each FFT bin E_i for $i = 0, 1, ...K_1$ is compared with a threshold th_1. We define Per as the percentage of the total number of bins where the energy is larger than th_1. If this percentage is larger than a limit p, $Per > p$, we assume the total band is too crowded to accommodate Cognitive Radio. If no bins have been found

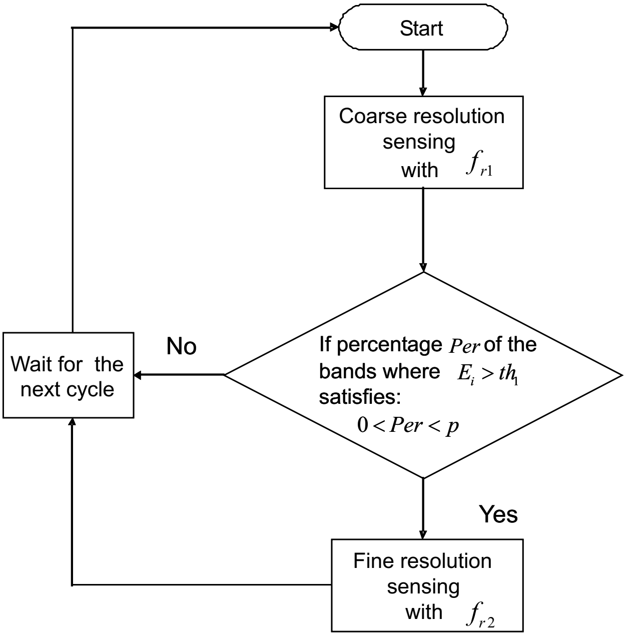
Start
Coarse resolution sensing with f_{r1}
If percentage Per of the bands where $E_i > th_1$ satisfies: $0 < Per < p$
No
Wait for the next cycle
Yes
Fine resolution sensing with f_{r2}

Figure 6.4: Flowchart of multi-resolution sensing

with significant energy (no i where $E_i > th_1$), namely $Per = 0$, we assume the band is empty. In these two conditions, fine resolution sensing is not needed and Cognitive Radio will either wait for a licensed user to free the spectrum or start communication. Otherwise, Cognitive Radio will continue with fine resolution sensing with a resolution of f_{r2} to focus on those high energy bands (like the colored potion in figure 6.3) where licensed users are potentially active. However, the specific method to select the interested bands is not considered in our discussion. The interesting portion of the spectrum is actually a part of the output bins of a larger K_2 size FFT, where $K_2 = \frac{B_{tot}}{f_{r2}}$. Based on the result of fine resolution sensing, Cognitive Radio will determine the transmission scheme and wait for the next sensing cycle. A flowchart describing the multi-resolution sensing scheme is shown in figure 6.4. The total computational cost of multi-resolution sensing can be expressed as:

$$C_{tot} = \{ \begin{array}{c} C_{coarse} + C_{fine} \text{ if } 0 < Per < p \\ C_{coarse} \text{ others} \end{array} \tag{6.11}$$

, where C denotes costs.

An important observation can be made for this multi-resolution sensing scheme: only a portion of the larger FFT outputs is needed for fine resolution sensing. In this case, the naive implementation of a larger size FFT is inefficient while an efficient algorithm which produces only the required parts of the FFT outputs, is desirable. The same observation has been made for the OFDM based Cognitive Radio with a large number of deactivated subcarriers in chapter 3. Therefore, the sparse FFT for OFDM based Cognitive Radio presented in chapter 4 perfectly suits the proposed multi-resolution sensing scheme and offers an efficient solution for fine resolution sensing. The details of the sparse FFT algorithm are omitted here, but can be found in chapter 4. We summarize its benefits in the context of multi-resolution sensing:

- it can produce a portion of FFT outputs in any position with a reduced computational complexity;
- it offers easy reconfigurability from a normal radix-2 FFT to a sparse FFT and vice versa, which minimizes the overhead due to the reconfiguration;
- its regular computational structure and easy reconfigurability facilitate hardware implementations;
- the core of the reconfigurable sparse FFT can be used for both OFDM modulation and spectrum sensing.

If the sparse FFT is used for fine sensing and only power-of-two size FFTs are considered, the cost of multi-resolution sensing in eq. 6.11 can be rewritten as:

$$C_{tot} = \{ \begin{array}{ll} \frac{K_1}{2}\log_2 K_1 + (N_2 - 1) * L + \frac{K_2}{2}\log_2 N_1 & \text{if } 0 < Per < p \\ \frac{K_1}{2}\log_2 K_1 & \text{others} \end{array} \tag{6.12}$$

K_1 and K_2 are the size of FFT for coarse sensing and fine sensing respectively. L denotes the number of outputs needed for fine resolution sensing of the K_2 size FFT. We choose N_1 as the nearest power-of-two integer larger than L and as a factor of K_2, denoted as $N_1 = \lceil L \rceil_{pow2}$. N_2 satisfies $N_1 N_2 = K_2$.

Let us consider a concrete multi-resolution sensing example. First, a coarse resolution sensing is done by a 128 point FFT with a resolution of $\frac{B_{tot}}{128}$. Suppose Cognitive Radio finds 5% of the total bandwidth needs fine resolution sensing, the system is reconfigured to a sparse 2k point FFT with the resolution of $\frac{B_{tot}}{2048}$ (16 times finer) to only focus on those bands of interest.

In this case, the total cost of multi-resolution sensing is the complexity of the sparse FFT-2048 with 0.05 nonzero ratio plus the complexity of the radix-2 FFT-128, totally about 9000 complex multiplications. The fixed resolution sensing using a radix-2 FFT-2048 requires 11264 ($\frac{2048}{2} \times \log_2 2048$) complex multiplications. Multi-resolution sensing thus gives about 20% saving in this particular example. However, this saving will diminish with the increasing percentage of the bandwidth which requires fine resolution sensing. The break even point in our example is about 25% of the total bandwidth which requires fine resolution sensing. Beyond this point, the complexity of the radix-2 FFT-128 plus sparse FFT-2048 will exceed the complexity of the radix-2 FFT-2048. Clearly, the condition to apply the proposed multi-resolution sensing method is that only a small fraction of total bandwidth requires fine resolution sensing.

6.2.3 Filter Bank Spectrum Sensing

The rectangular windowed FFT in OFDM based Cognitive Radio results in considerable interference to the licensed system due to high spectral leakage. The large spectrum leakage of the FFT also has a negative effect on the sensing reliability by reducing spectral dynamic range. The terminology, spectral dynamic range, here refers to the ratio between maximum and minimum spectral powers detected by energy detector. Higher spectral dynamic ranges can enhance detection probability. To reduce the spectrum leakage, nonrectangular windows can be applied to improve the dynamic range. Intuitively, windowing time domain samples has the same effect as filtering. The windowed FFT can be interpreted and implemented as an analysis filter bank [96]. Here, we refer to this filter bank based implementation of spectrum sensing as filter bank spectrum sensing. There is a growing interest on filter bank spectrum sensing for Cognitive Radio largely due to its easy integration with filter bank multicarrier systems [26, 77].

Figure 6.5 shows the block diagram of the filter bank spectrum sensing system. The signal after the ADC feeds into an M band analysis filter bank for energy estimation and followed by a threshold decision. The realization of the polyphase filter bank is based on the prototype filter $H(z)$ with length L [96]. $E_i(z)$ denotes the ith polyphase component of $H(z)$ and can be expressed as:

$$E_i(z) = h(i) + h(i+M)z^{-1} + ... + h(i+(K-1)M)z^{-(K-1)} \tag{6.13}$$

, where $h(i)$ denotes the ith coefficients of the prototype filter $H(z)$. K satisfies $K \times M = L$ and can be viewed as the number of taps for the

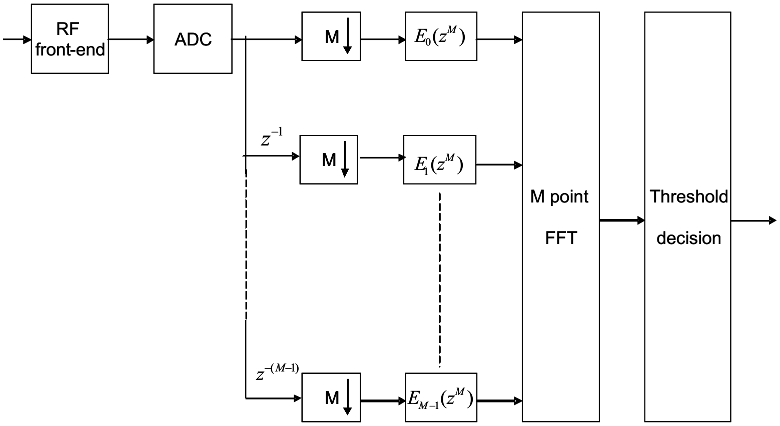

Figure 6.5: Block diagram of filter bank spectrum sensing

polyphase subfilters. The inputs to each polyphase components are downsampled substreams of the samples with an M times lower rate. In practice, the downsampling is implemented as a demultiplexer which delivers the samples to the polyphase component one after another. A time domain window is equivalent to a lowerpass FIR filter. If the length of the prototype filter $L = M$ and the coefficients are all set to one, we have the basic rectangular windowed FFT implementation. If we multiply a nonrectangular window with M FFT inputs, this is equivalent to applying a one tap filter on each polyphase component in the filter bank implementation. In [94], the author indicates that the window size should not be restricted by the FFT size in windowed FFT spectrum estimation. The length of the window L can be a multiple of the FFT size M, let $L = KM$. In the filter bank implementation, this case can be viewed as applying a K tap filter on each polyphase component [94]. The K tap filter on each polyphase component actually introduces a weighted averaging over K segments of M samples. This is similar to the so called method of weighted overlapped segment averaging [105]. In fact, the filter bank implementation model in figure 6.5 captures various windowed FFT based methods. Designing different windows is equivalent to designing different prototype filters to achieve good frequency resolution while reducing the sideband power.

We show an example of filter bank spectrum sensing for the detection of a narrowband signal. The size of the FFT M is chosen as 128 and the prototype filter is based on the Hamming window with the length $L = 128 \times 6$.

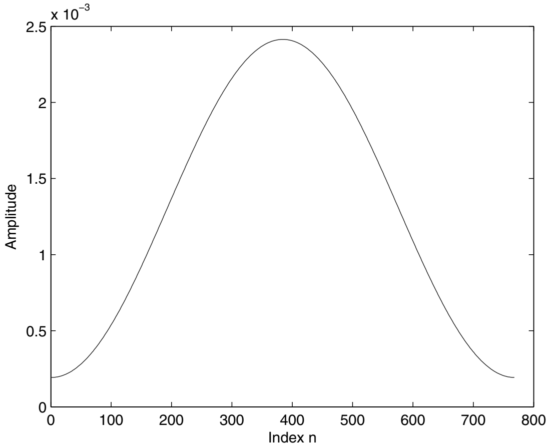

Figure 6.6: An example of impulse response of a prototype filter

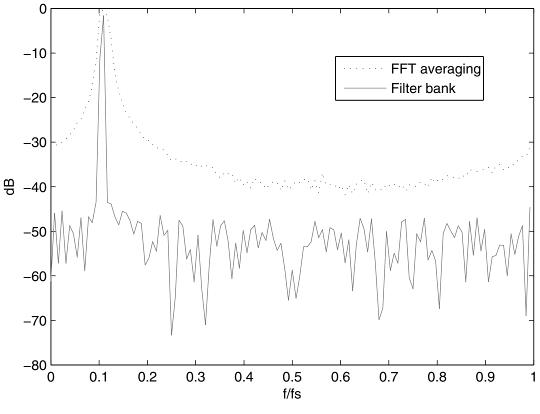

Figure 6.7: Spectrum estimation of filter bank and FFT averaging

So the coefficient of the prototype filter $h(i) = w(i)$, where $w(i)$ denotes the window coefficient and $i = 0, 1, ...128 \times 6 - 1$. Thus, the polyphase subfilter has $K = 6$ taps. The impulse response of the prototype filter is shown in figure 6.6. The implementation of the polyphase filter bank for spectrum sensing can be based on the GDFT implementation in section 5.3, but for a critically sampled case with no phase shift. The normalized spectrum estimation produced by the filter bank sensing is shown in figure 6.7. The targeted signal is a narrowband signal with 30dB SNR. The spectrum estimation of the filter bank is compared with the estimation produced by averaging 6 segments of the rectangular windowed FFT-128, plotted in the dotted line. Figure 6.7 shows that the filter bank sensing gives 20dB improvement in the dynamic range compared to simple FFT averaging. The total implementation costs of the filter bank includes the multiplication with the filter coefficients and one FFT. In this example, the costs of the filter bank sensing are 1216 ($128 \times 6 + \frac{128}{2} \log_2 128$) multiplications, 63% of which are for the filtering and 36% for the FFT.

From this example, we may conclude that filter bank based sensing can improve the sensing quality by designing a proper prototype filter. A method to design an optimal prototype filter, known as prolate filter, is introduced in [26]. The result in [26] is comparable to one of the best methods, namely Thomsons multitaper, for spectrum sensing. Another advantage of filter bank spectrum sensing is the easy integration with filter bank multicarrier systems.

6.3 Feature Detection

Due to multipath fading, signal attenuation and the performance of the radio front-end, the signal strength of the received licensed signal can be very low. One of the challenges for spectrum sensing is to detect the licensed signal buried in noise. As indicated in section 6.2, to detect the signal with a low SNR, a large number of samples has to be processed for energy detection. If the noise level is uncertain, energy detection will eventually become impossible below a certain SNR level. Therefore, in such a scenario, exploiting signal features for detection, also known as feature detection, could be a better solution. The rationale behind feature detection is that signals in general have inherent periodicity due to various modulation schemes while the noise has no such property. In the context of Cognitive Radio, a feature based detection method called cyclostationary feature detection has drawn much attention in the Cognitive Radio research community [10, 15, 46]. The cyclostationary feature detection is also considered as a spectrum sensing option in the AAF Cognitive Radio system.

6.3.1 Cyclostationary Feature Detection

Although interest in the cyclostationary feature detection in the context of Cognitive Radio has just started to grow, the theory on cyclostationary signals has been well studied and presented extensively in literature [30, 32, 33]. The motivation of defining a cyclostationary signal is to characterize and exploit second-order periodicity of a signal. A signal contains the second-order periodicity if the PSD of the product of the signal with its delayed version has spectral components at nonzero frequency. The fundamental parameter of second-order periodicity is the cyclic autocorrelation $R_x^\alpha(\tau)$, for a signal $x(t)$ which is defined as:

$$R_x^\alpha(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} (x(t+\tau/2)x^*(t-\tau/2)e^{-j2\pi\alpha t})dt \tag{6.14}$$

, where α is defined as the cycle frequency and τ is the delay. If $R_x^\alpha(\tau) \neq 0$ for some $\alpha \neq 0$, the signal $x(t)$ is considered as cyclostationary. An equivalent interpretation of a cyclostationary signal is that its autocorrelation function is periodic in time. By taking the Fourier transform on the cyclic autocorrelation function $R_x^\alpha(\tau)$, the other parameter, spectral correlation function (SCF) $S_x^\alpha(f)$, is derived as:

$$S_x^\alpha(f) = \int_{-\infty}^{\infty} R_x^\alpha(\tau)e^{-j2\pi f\tau}d\tau \tag{6.15}$$

The spectral correlation function $S_x^\alpha(f)$ is also known as the cyclic spectral density (CSD). The CSD is a two dimensional spectrum across the frequency f and the cycle frequency α which indicates the period of the signal. The PSD function is a special case of the CSD for $\alpha = 0$. The advantage of the CSD compared to the PSD for signal detection is that features of periodic signals with nonzero cycle frequencies can be exploited. Signals with different modulation types exhibit unique features at nonzero cycle frequencies while the noise has no features. If the licensed user's CSD is known, an optimal identification can be done in theory.

A detailed analysis on the cyclostationary feature detection is given in [32], followed by a conclusion that the cyclostationary feature detection generally outperforms energy detection for detecting periodical weak signals. As a result, the option of cyclostationary feature detection for spectrum sensing is justified. However, this conclusion may be overoptimistic when considering the reality of spectrum sensing for Cognitive Radio.

First of all, the cyclostationary feature detector requires some prior knowledge of the targeted signal. The cyclostationary detector presented in [32] is a single cycle detector which assumes prior knowledge of the signal's cycle frequency. However, in the Cognitive Radio scenario the licensed signal's cycle frequency may not be available. If a multicycle detector is applied, its implementation is considered to be difficult since the absolute phase of the signal must be known (or estimated) [32].

Secondly, various factors in reality may destroy or have negative effects on the cyclic feature of the signal. For example, the SCD is sensitive to the timing and phase of the signal [33]. This fact has been recognized by an experimental study where the sampling clock offset results in attenuated features [12]. The filtering, always involved in communication system, can also attenuate the cyclic features. For example, a pulse shaping filter with different roll-off factors β results in a large differences between the SCF for the same signal [12]. Moreover, the wireless channel may introduce various distortions which result in destruction of the feature.

Thirdly, in a real implementation, only finite discrete samples are used to estimate the CSD. The quality of estimation may vary a lot depending on the number of samples involved for processing. There exists spectral leakage and cycle leakage in the estimation [31]. To guarantee reliable estimation, a larger observation time and a finer spectral resolution are desirable. The condition for reliable sensing is that the product of the spectral resolution Δf and the temporal resolution Δt must greatly exceed unity [32], expressed as:

$$\Delta t \Delta f \gg 1 \tag{6.16}$$

Finally, cyclostationary feature detection is very computationally complex. The digital implementation of spectral correlation analyzers has been well studied in [7]. Two computationally efficient algorithms for cyclic spectral analysis have been suggested in [74], namely the FFT accumulation and the strip spectral correlation. Both methods involve FFTs and cross products, thus the computational complexity is still very high. An estimation of the cyclic spectrum with a reasonable resolution for a complex signal by applying the FFT accumulation method requires 73×10^6 complex multiplications according to [74]. The parallel description of the proposed methods in [74] is a good indication for mapping such complex algorithms onto architectures with multiprocessing units. As an effort to investigate the feasibility of cyclostationary feature detection in the context of the AAF project, Kokkeler et. al. have mapped the algorithm to calculate a 127×127 discrete SCF on a tiled SoC with 4 Montium cores [47]. Although the result shows the promise to process cyclostationary feature detection on a multi-core platform, the sensing time for the cyclostationary feature detector is still more than 10 times longer than its energy detection counterpart based on the FFT. The longer sensing time of the cyclostationary feature detection makes Cognitive Radio slow to adapt its transmission scheme. In some cases, cyclostationary feature detection may be too slow to capture changes in the spectrum usage pattern. Moreover, the computations involved with cyclostationary feature detection consume considerable power, especially when it has to be done frequently. Such a high power budget is a limiting factor for battery powered mobile radios.

Besides being a method for spectrum sensing, cyclostationary feature detection has another interesting application in the context of Cognitive Radio: self-identification, a method which can be used to establish the common control channel. Self-identification of an OFDM based Cognitive Radio can be realized by exploiting inherent cyclostationarity in the OFDM signal [110]. This feature occurs at the cycle frequency $a = \frac{k}{T}$, where T is the symbol time and k is the an integer multiple. However, this inherent cyclostationarity may not be suitable in the context of Cognitive Radio as argued by Sutton et. al. in [89]. They think the inherent cyclostationarity in the OFDM signal is rather rigid and the power of the features is low relative to that of the signal. Instead of exploiting inherent cyclostationarity in the OFDM signal, a distinctive cyclostationary signature is introduced for Cognitive Radio network identification [89]. The cyclostationary signature is created by transmitting redundant symbols on more than one subcarrier. In this way, a spectral correlation pattern can be found. By successfully detecting the unique cyclostationary signature, Cognitive Radio will identify its own network and establish a communication link. The authors in [89] also consider the case where the cyclostationary feature is destructed by frequency selective fading channel and propose to employ a multi-cycle feature signature as a counter measure.

In conclusion, the benefits of cyclostationary feature detection for Cognitive Radio are limited in reality. Therefore, it will only be considered as a complementary option for energy detection in bad SNR scenarios. However, the cyclostationary signature scheme proposed in [89] can be a promising solution for network coordination of Cognitive Radio.

6.3.2 Covariance Detection

It is worth mentioning another feature detection method based on the statistical covariance of the received signal [111]. This method is also considered by the IEEE 802.22 standard as a spectrum sensing option to detect a DTV signal. The covariance detection is based on the fact that the statistical covariance matrices or autocorrelations of signal and noise are generally different. The covariance matrix of white Gaussian noise R_n is always a diagonal matrix, written as:

$$R_n = \sigma_n^2 I \tag{6.17}$$

where I is the identity matrix and σ_n^2 is the noise variance. The apparent reason is that individual samples of the noise are considered to be uncorrelated. The covariance of the received signal R_x can be expressed as:

$$R_x = R_s + \sigma_n^2 I \tag{6.18}$$

where R_s is the covariance of the noise free signal. If no signal is present, the off-diagonal elements of R_x should be all zeros. If there are nonzero off-diagonal elements in R_x, a signal is considered to be present since the nonzero off-diagonal elements indicates inherent correlation which is a typical feature of signal.

Covariance detection assumes no prior knowledge of the signal and even the noise power. The detection will not be affected by linear distortions introduced by the channel and filters. The implementation of covariance detection based on finite samples can be found in [111]. The simulation results of covariance detection for the DTV signal show that it outperforms energy detection in cases where the noise levels are uncertain. However, it performs worse than energy detection when the noise level is known. The computational complexity of the covariance detection implementation is comparable to that of energy detection [111]. Therefore, it is an interesting alternative as a spectrum sensing scheme for Cognitive Radio.

6.4 Mapping onto the Reconfigurable Platform

Based on the previous discussions, we consider energy detection as the primary physical layer sensing scheme and cyclostationary feature detection method as a complementary scheme. In this section, we focus on mapping these spectrum sensing algorithms onto the reconfigurable platform. Since these sensing algorithms all involve a large amount of computations, the Montium is considered as the candidate to support these computationally intensive algorithms in an energy efficiency manner. Moreover, the Montium supports a flexible sensing system with its reconfigurability.

6.4.1 Energy Detection

It is beneficial to use the same algorithm for both sensing and transmission in terms of hardware reuse. Moreover, it also saves the efforts to develop new algorithms. The FFT algorithm can be used for both energy detection and OFDM transmission. The filter bank spectrum sensing can be easily integrated with filter bank multicarrier transmission. Mapping these algorithms onto the Montium has been treated in previous chapters. Here, we only make some emphasis in the context of sensing.

FFT Based Sensing on the Montium

Implementation of the FFT based sensing with a fixed resolution on the Montium is realized by using the existing FFT module in [39]. If the multi-resolution sensing scheme proposed in [118] is applied, a dynamic reconfigurable FFT module is required. We have presented an implementation of such a dynamically reconfigurable FFT module on the Montium in section 4.4 for OFDM based Cognitive Radio. This dynamically reconfigurable FFT module can reconfigure the FFT size and switch between a radix-2 FFT and a sparse FFT with a small reconfiguration overhead. As indicated in [118], this implementation can be efficiently reused for multi-resolution sensing.

Filter Bank Sensing on the Montium

The filter bank spectrum sensing system is actually a critically sampled analysis filter bank. Compared with the oversampled filter bank where phase shift has to be introduced for the FFT and no regular FIR exists, its implementation is more straightforward. If we consider the number of samples to be processed is a power-of-two integer M, a critically sampled analysis filter bank can be implemented as M FIR filters with the same number of taps K and one M point FFT as shown in figure 6.8. The coefficients for ith FIR filter can be expressed as $E_i = [h(i),\ h(i+M), ..., h(i+(K-1)M)]$, where $h(i)$ denotes the ith coefficient of the prototype filter. The length of the prototype filter $L = K \times M$.

Mapping a FIR filter onto the Montium has been well analyzed in [39]. For FIR filters with small number of taps, the Montium registers can store all the filter coefficients. For FIR filters with a large number of taps, the

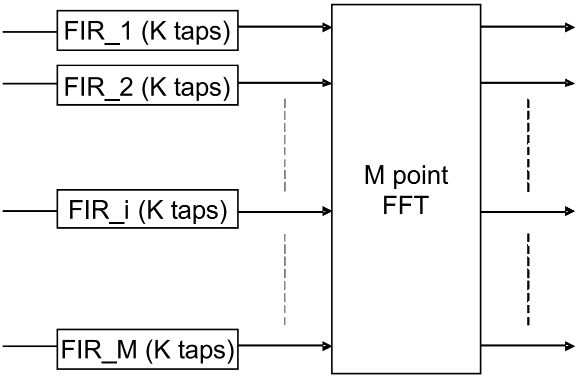

Figure 6.8: The critically sampled analysis filter bank for spectrum sensing

filter coefficients will be stored in the local memory of the Montium. The delay of the streaming filter is $\frac{K}{5}+3$ clock cycles for a FIR filter with K taps on the Montium [39].

However, we still need to make the mapping choices of M FIR filters. One choice is to map M FIR filters onto one Montium in a sequential fashion. The FIR filters are performed one after another from the first to the last. After all M samples are streamed out from the filters, the M point FFT is performed. If the memory based mapping scheme in [39] is considered, the delay of M FIR filters will be $M(\frac{K}{5}+3)$. If the other Montium is used to perform the FFT, it takes $M(\frac{K}{5}+3)+(\frac{M}{2}+2)\log_2 M$ clock cycles in total on two Montiums for the filter bank sensing. For the memory addressing it is required that $\frac{K}{5}$ is a power of two [39]. This can easily be achieved by adding coefficients of value zero to an FIR filter. Mapping M FIR filters with K taps onto one Montium is restricted by the size of the Montium memory. All L coefficients from M FIR filters should fit into 5 Montium local memories.

Instead of using one Montium to perform M FIR filters sequentially, we may considered to use a number of Montiums for FIR filtering in parallel (see figure 6.9). Let's consider using n Montiums for processing FIR filtering in parallel, where $ln = M$ and l is an integer. Each of the n Montiums performs l FIR filters sequentially, the filtering process takes $l(\frac{K}{5}+3)$ clock cycles to produce M input samples for the FFT. Therefore, the filter bank sensing can be done on $n+1$ Montiums (n Montiums for filtering and one for FFT) with $l(\frac{K}{5}+3)+(\frac{M}{2}+2)\log_2 M$ clock cycles. This parallel processing results

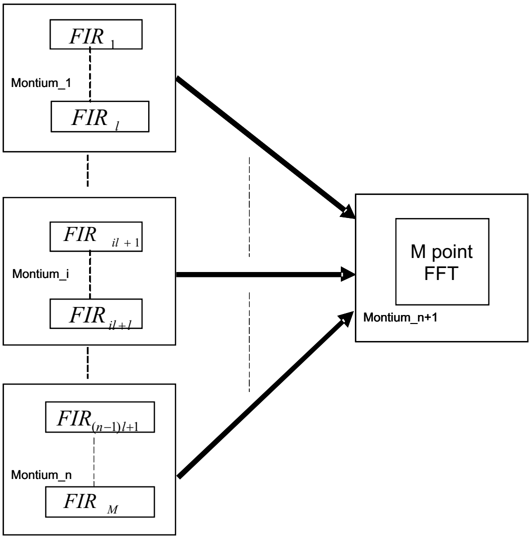

Figure 6.9: A parallel implementation of the filter bank spectrum sensing on $n+1$ Montiums

in a speedup R, which can be expressed as:

$$R = \frac{M(\frac{K}{5}+3) + (\frac{M}{2}+2)\log_2 M}{l(\frac{K}{5}+3) + (\frac{M}{2}+2)\log_2 M} \tag{6.19}$$

However, more Montiums are used in this parallel implementation. Thus the area is increased by a factor of $\frac{n+1}{2}$ to achieve the speedup R. Designers can make the tradeoff between area and speedup to meet the requirements of sensing time on a Montium based multi-core platform.

6.4.2 Cyclostationary Feature Detection

As we discussed in section 6.3, one of challenges of cyclostationary feature detection is its high computational complexity. The Montium is considered to support cyclostationary feature detection in [47]. Cyclostationary feature detection consists of a combination of an energy detector and a single correlator block as shown in figure 6.10. The main task for a cyclostationary feature detector is to determine the discrete spectral correlation function (DSCF) S_f^α, expressed as:

$$S_f^\alpha = \frac{1}{N}\sum_{n=0}^{N-1} X_{n,f+\alpha} \cdot X_{n,f-\alpha}^* \tag{6.20}$$

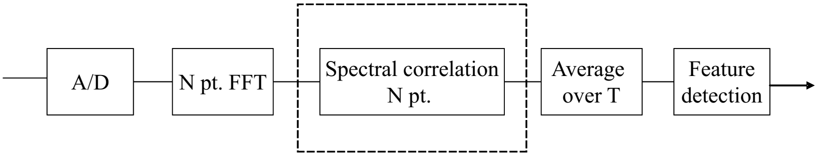

Figure 6.10: A cyclostationary feature detector

Task	#cycles
multiply accumulate	12192
read data	381
FFT	1040
reshuffling	256
initialisation	127
total	13996

Table 6.1: Number of cycles on the Montium to determine DSCF [47]

$X_{n,v}$ denotes the DFT, where f_s is the sampling frequency.

$$X_{n,v} = \sum_{k=0}^{K-1} x_{n+k} \cdot e^{j2\pi \frac{n+k}{f_s} v} \tag{6.21}$$

According to eq. 6.20, the complexity of cyclostationary feature detection is mainly due to $\frac{N^2}{4}$ complex multiplications involved in DSCF ($\frac{N}{2} \times \frac{N}{2}$ points). For example, calculating the DSCF for a 256 point spectrum involves 16 times as many complex multiplications as the determination of the spectrum itself [47]. As a design example, a 127×127 DSCF is mapped onto 4 Montium cores. In the first step, the tasks to be executed by each core are determined in a structured way using techniques known from the design of array processors. In the second step, the implementation of tasks on a processing core is analyzed. These implemented tasks are simulated for a single Montium tile with the Montium simulator. Table 6.1 gives an overview of the number of processor cycles required for the different tasks. The simulation result shows that the multiply and accumulate task for correlation is the most computationally complex task. It takes about 87% of the total cycles. The maximum clock speed of a Montium core equals 100 MHz and therefore the time required for the calculation of one integration step in the calculation of the 127×127 DSCF equals 139.96 μs [47]. However, a 256 point FFT computation only takes 10.4 μs at the maximum clock speed. Although the implementation of cyclostationary feature detection on the Montium based platform is efficient, it is 13 times slower than the FFT based energy detection. If cyclostationary feature detection does not improve the detection significantly, it is hard to justify applying such a computationally expensive algorithm.

The discussion in section 6.3 indicates that only under certain circumstances cyclostationary detection outperforms energy detection significantly (e.g. low SNR and noise uncertainty). Ideally, the sensing unit of Cognitive Radio can perform cyclostationary feature detection in these scenarios. Therefore, an adaptive sensing unit which has the capability to switch between energy detection and feature detection is more desirable. The combination of FFT based energy detection and cyclostationary feature detection is a viable option because the computation of DSCF includes an FFT. The reconfiguration from the FFT based energy detection to cyclostationary feature detection can be done by adding the spectral correlation task after the FFT computation as indicated in figure 6.10. Considering a Montium based multicore platform, one Montium is always active to perform the FFT based energy detection. When Cognitive Radio requires cyclostationary feature detection, the spectral correlation task can be mapped onto one or more inactive Montiums on the platform. A run-time mapping system proposed in [83] provides an interesting solution to map such dynamic applications onto a multiprocessor platform.

6.5 Chapter Summary

In this chapter, we discussed signal processing techniques for physical layer spectrum sensing. Two categories of sensing schemes have been considered, namely energy detection and feature detection.

Energy detection in general involves two steps: estimating the energy distribution of a signal in the frequency domain and making a threshold decision based on the estimate. Power spectral density estimation is often used to characterize the energy distribution of a signal in the frequency domain. A basic power spectrum estimation, also known as periodogram, can be obtained via the FFT algorithm. This FFT based spectrum sensing scheme can generate fast spectrum estimation and is easy to integrate with the OFDM based transmission scheme. The size of the FFT is a critical parameter because larger FFT sizes improve the bandwidth resolution and the processing gain but increase the sensing time. A variable size FFT enables multi-resolution sensing which avoids sensing the whole spectrum with maximum resolution all the time. As one of contributions of this thesis, a multi-resolution sensing scheme based on the sparse FFT has been presented in section 6.2.2. The proposed multi-resolution scheme is based on the observation that Cognitive Radio is only interested in a small portion of the spectrum with a finer resolution. This implies that only a small portion of the FFT outputs are needed for fine sensing. Therefore, the sparse FFT presented in chapter 4 suits this context quite well.

The energy detection based on the rectangular windowed FFT results in a reduced dynamic range due to large leakage power. As a result, time samples are often multiplied with a non-rectangular window before the FFT. Intuitively, time domain windowing is equivalent to frequency domain filtering. Therefore, various windowed FFT methods can be interpreted and implemented as an analysis filter bank. We have presented this filter bank based sensing scheme in section 6.2.3. The major advantage of filter bank spectrum sensing is its easy integration with the filter bank based multicarrier transmission scheme discussed in section 5.

Cyclostationary feature detection has been given much attention by the recent research on spectrum sensing for Cognitive Radio. We briefly introduce the classical cyclostationary theory and consider its advantages and disadvantages as a spectrum sensing scheme in the context of Cognitive Radio in section 6.3.1. The discussion on cyclostationary feature detection leads to a conclusion that, practically, it is not an attractive option as the primary spectrum sensing scheme. A feature detection scheme based on the covariance matrix of signal and noise is mentioned in section 6.3.2. This covariance based method outperforms energy detection for noise uncertain scenarios with a moderate computational complexity.

We summarize the major characteristics of different physical layer sensing methods in Table 6.2.

From the discussion in this chapter, we make several recommendations for spectrum sensing of Cognitive Radio.

- FFT based spectrum sensing should be considered as the primary sensing scheme for Cognitive Radio. It can generate fast sensing results with minimum prior knowledge (only the noise level). Integration with the OFDM based Cognitive Radio is easy.

- Multi-resolution spectrum sensing based on the sparse FFT enables more efficient sensing by reducing the number of unnecessary computations.

Table 6.2: Summary of spectrum sensing methods

Methods	Category	Proposed usage	Computation	Limitations
radix-2 FFT	energy	basic sensing or coarse sensing	radix-2 FFT	performance degradation in low SNR and under noise uncertainty
sparse FFT	energy	fine sensing in multi-resolution sensing	sparse FFT	the same limitation as the radix-2 FFT and computational saving is restricted
filter bank	energy	robust sensing with better dynamic range	radix-2 FFT + filtering	the same limitation as the radix-2 FFT and computational complexity
cyclostationary	feature	in low SNR or under noise uncertainty	radix-2 FFT + spectral correlation	feature impairments and computational complexity
covariance	feature	under noise uncertainty	filtering + autocorrelation + transforming covariance matrix [111]	performance in known noise

- Filter bank spectrum sensing is a good option for filter bank based Cognitive Radio.
- Cyclostationary feature detection can be employed as a complementary option for energy detection, especially for low SNR and noise uncertain scenarios. Cyclostationary feature detection is more attractive as a technique for self-identification of a Cognitive Radio network.
- Covariance detection can be considered for the noise uncertain case.
- Above all, the physical layer signal processing techniques alone may not be sufficient to obtain accurate spectrum occupancy information. Therefore, for spectrum sensing network collaboration is indispensable.

Mapping physical layer spectrum sensing algorithms onto a Montium based platform has been considered. The dynamically reconfigurable FFT module for OFDM based Cognitive Radio presented in section 4 can be reused for the multi-resolution sensing scheme presented in section 6.2.2. Implementing filter bank spectrum sensing as a critically sampled analysis filter bank onto the Montium has been considered. Two tasks are involved in filter bank spectrum sensing: FIR filtering and FFT. We proposed two mapping schemes: using one Montium for all FIR filters sequentially and using multiple Montiums in parallel for FIR filtering. The tradeoff between speedup and area by applying different mapping schemes is discussed. As an effort within the AAF project, cyclostationary feature detection has been mapped onto the Montium based platform in [47]. The result shows cyclostationary feature detection based on a 127×127 DSCF takes more than 10 times longer compared to its energy detection counterpart based on a 256 point FFT. A combination of the FFT based energy detection and cyclostationary feature detection can be an interesting option.

Chapter 7

Conclusions

7.1 Research Achievements

Cognitive Radio has been recognized as a future trend of wireless communication and opens a new research area in wireless communication. Most research focusses on the intelligence on the higher layers and simulation studies of the physical layer. The work in this thesis addresses Cognitive Radio from a different perspective. In chapter 1, we stated that the main objective of this research is to design, validate and implement high performance, adaptive and efficient physical layer algorithms for Cognitive Radio on a reconfigurable platform. To reach this main objective, we established several sub-objectives. In this thesis, a number of contributions have been made to achieve these objectives. The research achievements of this thesis are the following:

- An MPSoC platform based on coarse grained reconfigurable hardware is considered to support the adaptive baseband processing of Cognitive Radio. We present a task transaction level (TTL) interface approach both for developing a Cognitive Radio application at system level and as an interface between the application and the proposed MPSoC platform. The TTL approach has been used to map adaptive physical layer algorithms (e.g. the sparse FFT) for Cognitive Radio onto the MPSoC platform. It allows verifying the system's functional behavior and provides profile information for complexity analysis.

- Two types of multicarrier transmission schemes in the context of Cognitive Radio, namely OFDM and filter bank multicarrier, have been investigated. An OFDM system based on subcarrier deactivation and adaptive bit loading has been presented. From a computational point of view, the FFT is the bottleneck of the OFDM system. However, due to the deactivation of subcarriers, there could be a large number of zero inputs/outputs for the IFFT/FFT in an OFDM based Cognitive Radio system. In this case, the normal radix-2 IFFT/FFT is inefficient due to the wasted operations on zeros. Therefore, we present a novel sparse FFT as an efficient option for OFDM based Cognitive Radio to reduce the system complexity in case a large number of subcarriers are deactivated. Theoretically, the proposed sparse FFT leads to computational savings when the number of deactivated subcarriers is more than half of the total number of subcarriers. As a core algorithm for OFDM based Cognitive Radio, a dynamically reconfigurable FFT module which enables the reconfiguration of the FFT size and the reconfiguration between sparse FFT and radix-2 FFT has been mapped onto a Montium based MPSoC platform. The reconfiguration overhead is small and it is about 10 times more efficient than loading a new configuration completely for every change. The sparse FFT gives considerable computational savings in case a large number of subcarriers are deactivated.

One of the challenges of OFDM based Cognitive Radio is the appearance of sidelobes which may cause potential interference to the licensed system. Several methods were proposed in literature to mitigate the interference such as deactivating more subcarriers adjacent to the licensed system or applying non-rectangular windows. However, none of them gave a satisfactory solution. Therefore, we present an oversampled filter bank multicarrier system as an alternative transmission scheme for Cognitive Radio. The presented filter bank multicarrier system largely reduces sidelobes to mitigate potential interference. The sideband rejection can be below -30dB if a proper prototype filter is applied. Due to the computational complexity, the proposed filter bank multicarrier system imposes stringent processing requirements on the platform. We analyzed the mapping of the filter bank multicarrier onto the Montium, a key element on the proposed platform. The initial mapping result shows the promise of the proposed MPSoC platform to support a filter bank multicarrier system.

- As an essential part of the Cognitive Radio physical layer, several spectrum sensing techniques have been studied. We present a multiresolution spectrum sensing scheme based on the sparse FFT which only produces the required FFT bins. Therefore, it can focus on a portion of the spectrum with higher resolution with reduced computational effort. Filter bank based energy detection has been considered as an alternative for FFT based energy detection on the proposed reconfigurable platform. Finally, we have conceived a sensing unit which combines FFT based energy detection and cyclostationary feature detection on a reconfigurable platform.

7.2 Lessons Learned

- There exists a tradeoff between system performance and computational complexity. This is quite common in any communication systems including Cognitive Radio. For example, a filter bank multicarrier system has better sideband rejection than OFDM, but it is computationally more complex than OFDM due to extra filtering. To understand this tradeoff is important for designing algorithms which achieve the required system performance with minimum complexity.

- Take advantage of special signal structures to develop efficient algorithms. Due to its novel physical layer processing schemes, Cognitive Radio may have special signal structures thus offer many opportunities to develop new efficient DSP algorithms. A good example is the sparse FFT presented in chapter 4.

- Knowledge on the hardware architecture of a specific platform always helps to make an efficient implementation of an algorithm on this platform. Implementing algorithms on the Montium always needs optimizations of the algorithms. Making these optimizations is impossible without knowing the hardware architecture of the Montium.

- High level modelling is useful but one should be careful with quantitative results. As we have presented in chapter 4, the performance of the sparse FFT on the Montium is worse than the TTL due to the address generation bottleneck on the Montium.

7.3 Future Research Directions

The work in this thesis illustrates the issues of Cognitive Radio physical layer design on an MPSoC platform. We try to bring Cognitive Radio from a novel idea closer to reality. However, much work still has to be done to realize Cognitive Radio. We indicate a few future research directions from physical layer and radio platform perspectives.

Physical Layer

- *Optimum transmission scheme for Cognitive Radio*: The optimum choice of the transmission scheme for Cognitive Radio is still an open question. Two guiding principles for designing the transmission scheme for Cognitive Radio are: optimally using the available (sometimes segmented) spectrum and minimizing the interference to licensed users. We have considered the existing multicarrier techniques such as OFDM and filter bank in the context of Cognitive Radio and discussed their challenges. In both OFDM and filter bank systems, FFT/IFFT with the fixed sinusoid basis is used for transforming samples between the frequency and time domain. The fixed transform basis is rather inflexible for adaptation in the frequency and time domain. A wavelet based multicarrier system offers more opportunities for adaptation by employing different wavelets. Therefore, we think wavelet based multicarrier is an interesting transmission scheme to be considered for Cognitive Radio.

- *Adaptive sensing scheme*: We have discussed different sensing schemes including their advantages and disadvantages in the context of Cognitive Radio. It would be beneficial to combine some of the schemes and adaptively use these schemes based on specific scenarios. For example, an adaptive sensing unit, which combines FFT based energy detection and cyclostationary feature detection, uses energy detection as the primary detection method and switches to feature detection in certain scenarios (e.g. low SNR). This adaptive sensing scheme also has to be supported by a reconfigurable platform.

- *Efficient DSP algorithms*: One of the motivations to propose the sparse FFT for OFDM based Cognitive Radio is that a special signal structure gives an opportunity to develop new algorithms to achieve more efficient computation. This lesson can be learned to develop other efficient DSP algorithms (e.g. equalization) by taking advantage of the special signal structure of Cognitive Radio.

- *Cross-layer design*: Although our work mainly focuses on the physical layer issues of Cognitive Radio, we recognize that cross-layer design is very important for Cognitive Radio. The interactions between layers promotes adaptability at all layers of Cognitive Radio. For example, spectrum sensing for Cognitive Radio is a cross-layer design issue. Collaboration of different sensing nodes can enhance the detection performance and ease the processing requirements of individual nodes.

Radio Platform

- *Design methodology*: There is a gap between algorithm development and platform implementation. The rate of inventing new algorithms in communication systems is very fast. However, most of the algorithms only end up in computer simulations. Only a few algorithms are really implemented in hardware and/or software. One important reason is the lack of easy-to-use tooling for algorithm designers to quickly implement new ideas on the targeted hardware. We present a system-level design method for a MPSoC platform aiming to close this gap. However, this is only a small step. There remains much to be done. Ideally, an algorithm designer only needs to write MATLAB like code and a smart compiler will automatically map the algorithm onto the targeted platform. However, today we are far away from this idea.

- *Customizing reconfigurable architectures*: Customizing a reconfigurable architecture to certain algorithms is very beneficial. For example, the size of the local memory of the Montium can be increased to support large size FFTs and filter banks.

- *Run-time mapping system*: Dynamic applications such as Cognitive Radio may change their computation/communication requirements at run-time. For example, cyclostationary feature detection can be reconfigured from the basic energy detection algorithm by adding the spectral correlation task after the FFT task. If computational resources in the platform are scarce, such a change in the task graph may require a new mapping scheme at run-time. Moreover, Cognitive Radio is part of a trend that in the future algorithms will become more and more adaptive. This means that more decisions have to be taken at run-time.

Bibliography

[1] A. Abnous, H. Zhang, M. Wan, G. Varghese, V. Prabhu and J. Rabaey, "The Pleiades architecture," *The Application of Programmable DSPs in Mobile Communications*, A. Gatherer and A. Auslander, Eds., Wiley, 2002, pp. 327-360.

[2] P. Amini, R. Kempter, R-R. Chen, L. Lin and B. Farhang-Boroujeny, "Filter bank multitone: A candidate for physical layer of Cognitive Radio," *SDR Forum Technical Conference*, USA, 2005.

[3] American National Standards Institute (ANSI), "Network and customer installation interfaces asymmetric digital subscriber line (ADSL) metallic interface," ANSI standard T1.413-1995.

[4] V. Baumgarten, G. Ehlers, F. May, A. Nuckel, M. Vorbach and M. Weinhardt, "PACT XPPA self-reconfigurable data processing architecture," *The Journal of Supercomputing*, pp. 167-184, vol. 26(2), 2003.

[5] S. Brandes, I. Cosovic and M. Schnell, "Reduction of out-of-band radiation in OFDM based overlay systems," *Proceedings of the first IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN2005)*, Baltimore, USA, Nov 2005.

[6] R.W. Brodersen, A. Wolisz, D. Cabric, S.M. Mishra and D. Willkomm, "*A Cognitive Radio Approach for Usage of Virtual Unlicensed Spectrum*," White Paper, Jul. 2004.

[7] W.A. Brown and H.H. Loomis, "Digital implementations of spectral correlation analyzers," *IEEE Trans. on Signal Processing*, Feb. 1993.

[8] I. Budiarjo, H. Nikookar, and L.P. Ligthart, "Combined spectrum pooling and adaptive bit loading for Cognitive Radio OFDM based system," *in Proc. 13th Annual Symposium of IEEE on Communications and Vehicular Technology (SCVT) in the Benelux*, Liege, Belgium, Nov. 2006.

[9] M.D. van de Burgwal, G.J.M. Smit, G.K. Rauwerda and P.M. Heysters, "Hydra: an energy-efficient and reconfigurable network interface," *Proceedings of*

the 2006 International Conference on Engineering of Reconfigurable Systems and Algorithms, USA, 2006.

[10] D. Cabric and S.M. Mishra, "Implementation issues in spectrum sensing for Cognitive Radios," *38th Annual Asilomar Conference on Signals, Systems and Computers*, USA, 2004.

[11] D. Cabric, R. Brodersen, "Physical layer design issues unique to Cognitive Radio systems," *16th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2005)*, Sep. 2005.

[12] D. Cabric, *Cognitive Radios: System Design Perspective.* PhD Thesis, UC Berkeley, 2007.

[13] R.W. Chang, "Synthesis of band-limited orthogonol signals for multi-channel data transmission," *Bell Systems Technical Journal*, 46:1775-1796, 1966.

[14] H. Chen, W. Gao and D.G. Daut, "Signature based spectrum sensing algorithms for IEEE 802.22 WRAN," *Proceedings of IEEE International Conference on Communications (ICC 2007)*, Glasgow, UK, Jun. 2007.

[15] H. Chen, W. Gao and D.G. Daut, "Spectrum sensing using cyclostationary properties and application to IEEE 802.22 WRAN," *IEEE Global Telecommunications Conference (GLOBECOM)*, USA, Nov. 2007.

[16] G. Cherubini, E. Eleftheriou, S. Olcer and J.M. Cioffi, "Filter bank modulation techniques for very high speed digital subscriber lines," *IEEE Commun. Mag.*, May 2000.

[17] P.S. Chow, J.M. Cioffi, and J.A.C. Bingham, "A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels," *IEEE Trans. Commun.*, vol. 43, pp. 773775, Feb./Mar./Apr. 1995.

[18] J.W. Cooley and J.W. Turkey, "An algorithm for the machine caculation of complex fourier series," *Mathematics of Computation*, Apr. 1965.

[19] R.E. Crochiere and L.R. Rabiner, *Multirate Digital Signal Processing.* Prentice-Hall, 1983.

[20] T.L. Doumi, "Spectrum considerations for public safety in the United States," *IEEE Commun. Mag.*, Jan. 2006.

[21] European Telecommunications Standards Institute (ETSI), *Broadband Radio Access Network (BRAN); HiperLAN Type 2; Physical (PHY) Layer*, ETSI TS 101 475 v1.2.2(2001-02), Feb. 2001.

[22] European Telecommunications Standards Institute (ETSI), *Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to Mobile, Portable and Fixed Receivers*, European Telecommunication Standard ETS 300 401, 1st edition, reference DE/JTC-DAB, February 1995.

[23] European Telecommunications Standards Institute (ETSI), *Digital Radio Mondiale (DRM); System Specification*, ETSI TS 101 980 v1.1.1 (2001-09), Sep. 2001.

[24] European Telecommunications Standards Institute (ETSI), *Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television*, European Telecommunications Standard, ETS 300 744 1st edition, reference DE/JTC-DVB-8, March 1997.

[25] B. Farhang-Boroujeny, "Multicarrier modulation with blind detection capability using cosine modulated filter banks," *IEEE Trans. Commun.*, Dec. 2003.

[26] B. Farhang-Boroujeny, "Filter bank spectrum sensing for Cognitive Radios," *IEEE Transactions on Signal Processing*, May, 2008.

[27] Federal Communication Commision, "Spectrum Policy Task Force, Tech. Report, Nov. 2002.

[28] R.F.H. Fischer and J.B. Huber, "A new loading algorithm for discrete multitone transmission," *Proc. IEEE Global Telecommun. Conf.*, vol. 1, pp. 724.728, 1996.

[29] Y. Guo, "*Mapping Applications to a Coarse-Grained Reconfigurable Architecture*", PhD Thesis, University of Twente, 2006

[30] W.A. Gardner, *Statistical Spectral Analysis: A Non-probabilistic Theory.* Prentice-Hall, 1987.

[31] W.A. Gardner, "Measurement of spectral correlation," *IEEE Trans. on Acoustics, Speech and Signal Processing*, Oct. 1986.

[32] W.A. Gardner, "Signal interception: A unifying theoretical framework for feature detection," *IEEE Trans. on Communications*, Aug. 1988.

[33] W.A. Gardner, "Exploitation of spectral redundancy in cyclostationary signals," *IEEE Signal Processing Magazine*, Apr. 1991.

[34] Marcel Hammer *Extending a Lightweight Real-Time Kernel*, Master Thesis, University of Twente, 2008.

[35] R. Hartenstein, "A decade of reconfigurable computing: a visionary retrospective," *Proceedings of the Conference Design Automation and Test in Europe*, Munich, Germany, 2001.

[36] M.H. Hayes, *Statical signal processing and modeling.* John Weily and Sons Inc, 1996.

[37] S. Haykin, "Cognitive Radio: brain-empowered wireless communication," IEEE J. Select. Areas Commun., Feb. 2005.

[38] S. He and M. Torkelson "Computing partial DFT for comb spectrum evaluation," *IEEE Signal Processing Letters*, Jun. 1996.

[39] P. Heysters, *Coarse-Grained Reconfigurable Processors; Flexibility Meets Efficiency*, PhD Thesis, University of Twente, Sep. 2004.

[40] F.W. Hoeksema, M. Heskamp, R. Schiphorst and C.H. Slump, "A node architecture for disaster relief networking," *Proceedings of the first IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN2005)*, Baltimore, USA, Nov 2005.

[41] Y. Hur, J. Park, W. Woo, K. Lim, C.H. Lee, H.S. Kim and J. Laskar, "A wideband analog multi-resolution spectrum sensing (MRSS) technique for cognitive radio (CR) systems," *International Symposium on Circuits and Systems (ISCAS 2006)*, USA, 2006.

[42] T. Ihalainen, J. Alhava, A. Viholainen, H. Xing, J. Rinne, and M. Renfors, "On the performance of filter bank based multicarier systems in xDSL and WLAN applications," in *Proc. IEEE Int. Conf. Commun.*, pp. 11201124, vol. 2, Jun. 2000.

[43] J. Kang, T. Henriksson and P. van der Wolf, "An interface for the design and implementation of dynamic applications on multi-processor architectures," *Proceedings of ESTImedia*, 2005.

[44] N.K. Kavaldjiev, *A run-time reconfigurable Network-on-Chip for streaming DSP applications*. PhD thesis, University of Twente, 2007.

[45] T. Keller and L. Hanzo, "Adaptive multicarrier modulation: a convenient framework for time-frequency processing in wireless communications," *Proc. IEEE*, vol. 88, pp. 611640, May 2000.

[46] K. Kim, I.A. Akbar, K.K. Bae, U. Jung-sun, C.M. Spooner and J.H. Reed, "Cyclostationary approaches to signal detection and classification in Cognitive Radio," *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 17-20 Apr. 2007.

[47] A.B.J. Kokkeler, G.J.M. Smit, T. Krol and J. Kuper, "Cyclostationary feature detection on a tiled-SoC," *Proceedings of DATE2007*, France, 2007.

[48] M.K. Lakshmanan, I. Budiarjo and H. Nikookar, "Maximally frequency selective wavelet packets based multi-carrier modulation scheme for Cognitive Radio systems," *IEEE GLOBECOM conference*, 2007.

[49] B. Le, P. Garcia, Q. Chen , B. Li , F. Ge, M. ElNainay , T.W. Rondeau and C.W. Bostian,"A public safety Cognitive Radio node system," *Software Defined Radio Technical Conference*, USA, Nov. 2007.

[50] L. Lin and B. Farhang-Boroujeny, “Cosine modulated multitone for very high-speed digital subscriber lines,” EURASIP Journal on Applied Signal Processing, vol 2006, Article ID 19329, 2006.

[51] J.D. Markel, “FFT Pruning,” *IEEE Trans. on Audio and Electroacoustics*, Dec. 1971.

[52] K.W. Martin, “Small side-lobe filter design for multitone data-communication applications,” *IEEE Trans. Circuit and System*, Aug. 1998

[53] M.A. McHenry, “NSF spectrum occupancy measurements project summary,” Shared Spectrum Company, Aug. 2005.

[54] M. McHenry, E. Livsics, T. Nguyen and N. Majumdar, “XG dynamic spectrum sharing field test results,” *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2007)*, 2007.

[55] S.M. Mishra, D. Cabric, C. Chang, D. Willkomm, B. van Schewick, A. Wolisz, R. Brodersen, “A real time Cognitive Radio testbed for physical and link layer experiments,” *IEEE DySPAN*, Nov. 2005.

[56] S.M. Mishra, A. Sahai, and R. Brodersen, ”Cooperative sensing among cognitive radios,” *International Conference on Communications*, 2006.

[57] J. Mitola, “Cognitive Radio: Making software radios more personal,” *IEEE Personal Communications*, vol 6, 1999.

[58] J. Mitola, *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*, PhD Thesis, Royal Institute of Technology, Sweden, May 2000.

[59] G.E. Moore, “Cramming more components onto integrated circuits,” *Electronics Magazine*, 38(8), April 9, 1965.

[60] R. van Nee and R. Prasad, *OFDM for Wireless Communication Systems.* Artech House, 2004.

[61] N.M. Neihart, S. Roy, and D.J. Allstot, “A parallel, multi-resolution sensing technique for multiple antenna Cognitive Radios,” *International Symposium on Circuits and Systems (ISCAS 2007)*, USA, 2007.

[62] H. Nikookar and M.K. Lakshmanan, “Comparison sensitivity of OFDM and wavelet packet modulation to time synchronization error,” *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2008)*, France, 2008.

[63] P. Pawelczak, R. Venkatesha Prasad, L. Xia and I.G.M.M. Niemegeers, “Cognitive Radio emergency networks-requirements and design,” *Proc. 1st IEEE Symposium on New Frontiers on Dynamic Spectrum Access Networks (IEEE DySPAN 2005)*, USA, Nov. 2005.

[64] T. Pollet, M. van Bladel, and M. Moeneclaey, "BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 191-193, Feb./Mar./Apr. 1995.

[65] J. G. Proakis, *Digital Communications.* McGraw-Hill, 3rd ed., 1995.

[66] R. Rajbanshi, A.M. Wyglinski, and G.J. Minden, "An efficient implementation of NC-OFDM transceivers for Cognitive Radios," *Proceedings of the 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, Mykonos Island, Greece, Jun. 2006.

[67] U. Ramacher, "Software defined radio prospects for multistandard mobile phones," *IEEE Transactions on Computer*, Oct. 2007.

[68] T.S. Rappaport, *Wireless Communications: Principles and Practice.* Prentice Hall, 1996.

[69] G.K. Rauwerda, G.J.M. Smit, C.R.W. van Benthem and P.M. Heysters, "Reconfigurable Turbo/Viterbi channel decoder in the coarse-grained Montium architecture," In: *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA06)*, 26-29 Jun. 2006.

[70] G.K. Rauwerda, P.M. Heysters and G.J.M. Smit, "Towards software defined radios using coarse-grained reconfigurable hardware," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Jan. 2008.

[71] W. Rhee, J.C. Chuang and L.J. Cimini, "Performance comparison of OFDM and multitone with polyphase filter bank for wireless communications," *Proc. Vehicular Technology Conference*, May. 1998.

[72] C.J. Rieser. *Biologically Inspired Cognitive Radio EngineModel Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless Communications and Networking*, PhD thesis, Virginia Tech., 2004.

[73] A. Rivaton, J. Quevremont, Q. Zhang, P.T. Wolkotte and G.J.M. Smit, "Implementing non power-of-two FFTs on coarse-grain reconfigurable architectures," In: *Proceedings of the International Symposium on System-on-Chip (SoC 2005)*, Nov. 2005.

[74] R.S. Roberts, W.A. Brown and H.H. Loomis, "Computationally efficient algorithms for cyclic spectral analysis," *IEEE Signal Processing Magazine*, Apr. 1991.

[75] F.W Seelig, "A Description of the August 2006 XG Demonstrations at Fort A.P. Hill," *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2007)*, 2007.

[76] S.F.A. Shah and A.H. Tewfik, “Efficient design of OFDMA-based programmable wireless radios,” *EURASIP Journal on Wireless Communications and Networking*, 2008.

[77] F. Sheikh and B. Bing, “Cognitive spectrum sensing and detection using polyphase DFT filter banks,” *IEEE Consumer Communications and Networking Conference, 2008 (CCNC 2008)*, 10-12 Jan. 2008.

[78] H. Singh, M.H. Lee, G. Lu, F. Kurdahi, N. Bagherzadeh and E. Chaves, “MorphoSys: an integrated reconfigurable system for data-parallel and computation intensive applications,” IEEE Trans. Computers, 49 (5), 2000.

[79] Bas van Sisseren *Design of a Lightweight Real-Time Streaming Kernel*, Master Thesis, University of Twente, 2007.

[80] D. P. Skinner “Pruning the decimation In-Time FFT algorithm,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Apr. 1976.

[81] G.J.M. Smit, E. Schuler, J.E. Becker, J. Quevremont and W. Brugger, “Overview of the 4S project”, *International Symposium on System-on-Chip*, Nov 2005.

[82] G.J.M. Smit, A.B.J. Kokkeler, P.T. Wolkotte and M.D. van de Burgwal, “Multi-core architectures and streaming applications,” *Proceedings of the Tenth International Workshop on System-Level Interconnect Prediction (SLIP 2008)*, UK, 2008.

[83] L.T. Smit, G.J.M. Smit, J.L. Hurink, H.J. Broersma, D. Paulusma and P.T. Wolkotte, “Run-time mapping of applications to a heterogeneous reconfigurable tiled system on chip architecture,” *Proceedings of the International Conference on Field-Programmable Technology*, Australia, 2004.

[84] E. Sofer and G. Chouinard, “WRAN channel modelling,” IEEE 802.22-05/0055r3, Jul. 2005. available online: `https://mentor.ieee.org/802.22/file/05/22-05-0055-03-0000-wran-channel-modeling.doc`

[85] H.V. Sorensen and B. Sidney, “Efficient computation of the DFT with only a subset of input or output points,” *IEEE Trans. on Signal Processing*, Mar. 1993.

[86] T.V. Sreenivas and P.V.S. Rao “FFT algorithm for both input and output pruning,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Jun. 1979.

[87] S. Srikanteswara, R.C. Palat, J.H. Reed and P. Athanas, “An overview of configurable computing machines for software defined radio hansets,” *IEEE Communications Magazine*, Jul. 2003.

[88] G. Stitt, F. Vahid and S. Nematbakhsh, "Energy savings and speedups from partitioning critical software loops to hardware in embedded systems," *ACM Transaction on Embedded Computer System*, Mar. 2004.

[89] P.D. Sutton, K.E. Nolan and L.E. Doyle, "Cyclostationary signatures in practical Cognitive Radio applications," *IEEE Journal on Selected Areas in Communications*, Jan. 2008.

[90] Synopsis website: `http://www.synopsis.com`

[91] K. Takeuchi, T. Fukuhara, S. Nomura and S. Yamamoto, "Cognitive Radio using multi-transmission links," *Proceedings of 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2007)*, Orlando, USA, 31 Jul.-03 Aug. 2007.

[92] R. Tandra and Anant Sahai, "SNR walls for signal detection," *IEEE Journal on Special Topics in Signal Processing*, Feb 2008.

[93] T.J. Todman, G.A. Constantinides, S.J.E. Wilton, O. Mencer, W. Luk and P.Y.K. Cheung, "Reconfigurable computing: architectures and design methods," *IEE Proceedings Computers and Digital Techniques*, Mar. 2005.

[94] T. Thong, "Practical consideration for a continuous time digital spectrum analyzer," *Proc. IEEE International Symposium Circuits and Systems (ISCAS)*, May 1989.

[95] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, Apr. 1967.

[96] P.P. Vaidyanathan, *Multirate System and Filter Banks*. Prentice-Hall, 1993.

[97] M. Vetterli, "Perfect transmultiplexers," *Proceedings of ICASSP*, pp. 25672570, 1986.

[98] "Very-high-bit-rate digital subscriber line (VDSL) metallic interface Part 1,2 and 3," Committee T1, Work Group T1E1.4, Contribution T1E1.4/2000-009R3,-011,-013R4, Feb. 2001.

[99] A.J. Viterbi, "Wireless digital communication: a view based on three lessons learned," *IEEE Communications Magazine*, Sep. 1991.

[100] K.H.G. Walters, *Cognitive Radion on a reconfigurable platform*, Master's thesis, University. of Twente.

[101] S. Weistein and P. Ebert, "Data transmission by frequency division multiplexing using the discrete fourier transform," *IEEE Trans. Commun. Techn.*, vol. COM-19, pp 628-634, Oct. 1974.

[102] S. Weiss and R.W. Stewart, ”Fast implementation of oversampled modulated filter banks,” *IEE Electronics Letters*, Vol.36(No.17): pp.1502-1503, August 2000.

[103] T.A. Weiss and F.K. Jondral, ”Spectrum pooling: an innovative strategy for the enhancement of spectrum efficiency,” *IEEE Commun. Mag.*, Mar. 2004.

[104] T. Weiss, A. Krohn, J. Hillenbrand, and F. Jondral.“Mutual Interference in OFDM based Spectrum Pooling Systems,” *Proc of IEEE VTC*, Italy, May 2004.

[105] P.D. Welch, “The use of FFT for the estimation of power spectra: A method based on time averaging over short modified periodograms,” *IEEE Trans. Audio Electroacoust.*, Jun. 1967.

[106] P. van der Wolf, E.A. de Kock, T. Henriksson, W. Kruijtzer and G. Essink, “Design and programming of embedded multiprocessors: an interface-centric approach,” *Proceedings of ISSS+CODES*, Sep. 2004.

[107] P.T. Wolkotte, G.J.M. Smit, G.K. Rauwerda and L.T. Smit, “An energy-efficient reconfigurable circuit switched Network-on-Chip,” *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, USA, 2005.

[108] A.M. Wyglinski, F. Labeau, and P. Kabal, “An efficient bit allocation algorithm for multicarrier modulation,” in *Proc. IEEE Wireless Commun. Networking Conf.*, Atlanta, GA, USA, Mar. 2004.

[109] A.M. Wyglinski, “Effects of bit allocation on non-contiguous multicarrier-based Cognitive Radio transceivers,” *64th IEEE Vehicular Technology Conference*, Montreal, QC, Canada, Sep. 2006.

[110] T. Yucek and H. Arslan, “OFDM signal identification and transmission parameter estimation for Cognitive Radio applications,” *Proc. IEEE Global Telecommunication Conference (GLOBECOM07)*, USA, Nov. 2007.

[111] Y. Zeng and Y. Liang, “Covariance based signal detections for Cognitive Radio,” *Proceedings of the second IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN2007)*, Apr. 2007.

[112] Q. Zhang, A.B.J. Kokkeler and G.J.M. Smit, “A reconfigurable radio architecture for Cognitive Radio in emergency networks,” *Proceedings of European Conference on Wireless Technology (ECWT 2006)*, pp. 35-38, Manchester, UK, 10-15 Sep. 2006.

[113] Q. Zhang, A.B.J. Kokkeler, and G.J.M. Smit, “Adaptive OFDM system design for Cognitive Radio,” *Proceedings of 11th International OFDM-Workshop*, pp. 91-95, Hamburg, Germany, 30-31 Aug. 2006.

[114] Q. Zhang, A.B.J. Kokkeler and G.J.M. Smit, “An efficient FFT for OFDM based Cognitive Radio on a reconfigurable architecture,” *Proceedings of IEEE International Conference on Communications (ICC 2007)*, pp. 6522-6526, Glasgow, UK, 24-28 Jun 2007.

[115] Q. Zhang, A.B.J. Kokkeler and G.J.M. Smit, “Cognitive Radio design on an MPSoC reconfigurable platform,” *Proceedings of 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2007)*, Orlando, USA, 31 Jul.-03 Aug. 2007.

[116] Q. Zhang, A.B.J. Kokkeler and G.J.M. Smit, “A system-level design method for Cognitive Radio on a reconfigurable multi-processor architecture,” *Proceedings of International Symposium on System-on-Chip (SoC2007)*, pp. 3-6, Tampere, Finland, 19-21 Nov 2007.

[117] Q. Zhang, A.B.J. Kokkeler and G.J.M. Smit, “Dynamically reconfigurable FFTs for Cognitive Radio on a multiprocessor platform,” *International Conference on Engineering of Reconfigurable Systems and Algorithms 2008 (ERSA2008)*, USA, Jul. 2008.

[118] Q. Zhang, A.B.J. Kokkeler and G.J.M. Smit, “An efficient multi-resolution spectrum sensing method for Cognitive Radio,” *CHINACOM 2008 COGCOM workshop (COGCOM2008)*, Hangzhou, China, Aug. 2008

[119] Q. Zhang, A.B.J. Kokkeler and G.J.M. Smit, “An oversampled filter bank multicarrier system for Cognitive Radio,” *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2008 (PIMRC2008)*, France, Sep. 2008.

[120] W.Y. Zou and Y. Wu, “COFDM: an overview,” *IEEE Transactions on Broadcasting*, vol. 41, no.1, pp. 1-8, Mar. 1995.

Publications

[QZ1] A. Rivaton, J. Quevremont, **Q. Zhang**, P.T. Wolkotte and G.J.M. Smit, “Implementing non power-of-two FFTs on coarse-grain reconfigurable architectures,” *Proceedings of the International Symposium on System-on-Chip (SoC 2005)*, pp. 82-85, Tampere, Finland, 14-17 Nov. 2005. IEEE Computer Society. ISBN 0-7803-9294-9.

[QZ2] **Q. Zhang**, G.J.M. Smit and L.T. Smit, “A reconfigurable platform for Cognitive Radio,” *Proceedings of Second International Conference on Mobile Technology, Applications and Systems*, pp. 39-47, Guangzhou, China, 15-17 Nov. 2005. IEEE Computer Society. ISBN 981-05-4571-1.

[QZ3] **Q. Zhang**, F.W. Hoeksema, A.B.J. Kokkeler and G.J.M. Smit, “Towards Cognitive Radio for emergency networks,” In *Mobile Multimedia: Communication Engineering Perspective*, Nova Publishers, USA, 2006. ISBN 1-60021-207-7.

[QZ4] **Q. Zhang**, A.B.J. Kokkeler, and G.J.M. Smit, “Adaptive OFDM system design for Cognitive Radio,” *Proceedings of 11th International OFDM-Workshop*, pp. 91-95, Hamburg, Germany, 30-31 Aug. 2006. IEEE Communications Society, Germany Chapter.

[QZ5] **Q. Zhang**, A.B.J. Kokkeler and G.J.M. Smit, “A reconfigurable radio architecture for Cognitive Radio in emergency networks,” *Proceedings of European Conference on Wireless Technology (ECWT 2006)*, pp. 35-38, Manchester, UK, 10-15 Sep. 2006. IEEE Communication Society. ISBN 2-9600551-5-2.

[QZ6] **Q. Zhang**, A.B.J. Kokkeler and G.J.M. Smit, “Cognitive Radio for emergency networks,” *Proceedings of Global Mobile Congress 2006*, pp. 32-37, Beijing, China, 09-11 Oct. 2006. Delson Group. ISSN 1557-0622.

[QZ7] **Q. Zhang**, A.B.J. Kokkeler and G.J.M. Smit, "An efficient FFT for OFDM based Cognitive Radio on a reconfigurable architecture," *Proceedings of IEEE International Conference on Communications (ICC 2007)*, pp. 6522-6526, Glasgow, UK, 24-28 Jun 2007. IEEE Communications Society. ISBN 1-4244-0353-7.

[QZ8] **Q. Zhang**, A.B.J. Kokkeler and G.J.M. Smit, "Cognitive Radio design on an MPSoC reconfigurable platform," *Proceedings of 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2007)*, Orlando, USA, 31 Jul.-03 Aug. 2007. IEEE Communications Society. ISBN 1-4244-0815-6.

[QZ9] **Q. Zhang**, A.B.J. Kokkeler and G.J.M. Smit, "A system-level design method for Cognitive Radio on a reconfigurable multi-processor architecture," *Proceedings of International Symposium on System-on-Chip (SoC2007)*, pp. 3-6, Tampere, Finland, 19-21 Nov 2007. IEEE Circuits and Systems Society. ISBN 1-4244-1368-0.

[QZ10] **Q. Zhang**, A.B.J. Kokkeler and G.J.M. Smit, "Cognitive Radio design on an MPSoC reconfigurable platform," *ACM/Springer Journal of Mobile Networks and Applications*, 13(5), pp. 424-430, Oct. 2008. ISSN 1383-469X.

[QZ11] **Q. Zhang**, K.H.G. Walters, A.B.J. Kokkeler and G.J.M. Smit, "Dynamically reconfigurable FFTs for Cognitive Radio on a multiprocessor platform," *International Conference on Engineering of Reconfigurable Systems and Algorithms 2008 (ERSA2008)*, USA, Jul. 2008.

[QZ12] **Q. Zhang**, A.B.J. Kokkeler and G.J.M. Smit, "An efficient multi-resolution spectrum sensing method for Cognitive Radio," *CHINACOM 2008 COGCOM workshop (COGCOM2008)*, Hangzhou, China, Aug. 2008. ISBN 978-1-4244-2374-3

[QZ13] **Q. Zhang**, A.B.J. Kokkeler and G.J.M. Smit, "An oversampled filter bank multicarrier system for Cognitive Radio," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2008 (PIMRC2008)*, France, Sep. 2008. ISBN 978-1-4244-2644-7.

[QZ14] A.B.J. Kokkeler, G.K. Rauwerda, P.T. Wolkotte, **Q. Zhang**, P.K.F. Holzenspies and G.J.M. Smit, "Reconfigurable baseband processing for wireless communications," In Adaptive Signal Processing for Wireless Communications, CRC publishers, pp. 443-478, USA, 2009. ISBN 978-1-4200-4601-4

[QZ15] **Q. Zhang**, A.B.J. Kokkeler, G.J.M. Smit and K.H.G. Walters, “Cognitive Radio baseband processing on a reconfigurable platform,” Special issue on Cognitive Radio: Algorithms and System Design, Elsevier Journal of Physical Communication, 2009. (accepted for publication).

Appendix A

Mathematics on Oversampled Filter Bank

Here we give a more detailed mathematical derivation for the GDFT based oversampled filter bank implementation in section 5.3. Following eq. 5.6 in section 5.3, the synthesis filter bank operation can be expressed in a matrix form:

$$H_s = [p(k - \frac{L-1}{2})e^{j2\pi(m-\frac{M-1}{2})(k-\frac{L-1}{2})/M}]_{M\times L}$$

Then the $M \times L$ matrix $[H_s]_{M\times L}$ can be written as:

$$[H_s]_{M\times L} = [T]_{M\times M} \times [I_M \ (-1)^{M-1} I_M]_{M\times 2M} \times [I_{2M} \ I_{2M} ... \ I_{2M} \ \hat{I}_{2M,u}]_{2M\times L} \times [\Lambda_p]_{L\times L}$$

where

$$[T]_{M\times M} = \begin{bmatrix} e^{-j\pi(-\frac{M-1}{2})(L-1)/M} & & & & \\ & \ddots & & & \\ & & e^{-j\pi(i-\frac{M-1}{2})(L-1)/M} & & \\ & & & \ddots & \\ & & & & e^{-j\pi(\frac{M-1}{2})(L-1)/M} \end{bmatrix}_{M\times M}$$

$$\times \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{\frac{j2\pi}{M}} & \cdots & e^{\frac{j2\pi(M-1)}{M}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{\frac{j2\pi(M-1)}{M}} & \cdots & e^{\frac{j2\pi(M-1)(M-1)}{M}} \end{bmatrix}_{M\times M}$$

$$\times \begin{bmatrix} e^{0} & & & & \\ & \ddots & & & \\ & & e^{-j\pi i(M-1)/M} & & \\ & & & \ddots & \\ & & & & e^{-j\pi(M-1)(M-1)/M} \end{bmatrix}_{M\times M}$$

Appendix A

$$[I_M \ (-1)^{M-1} I_M]_{M\times 2M} = \begin{bmatrix} 1 & & & (-1)^{M-1} & & \\ & \ddots & & & \ddots & \\ & 1 & & & (-1)^{M-1} & \\ & & \ddots & & & \ddots \\ & & 1 & & & (-1)^{M-1} \end{bmatrix}_{M\times 2M}$$

$$[I_{2M} \ I_{2M} \cdots \ I_{2M} \ \hat{I}_{2M,u}]_{2M\times L} = \begin{bmatrix} 1 & & & \cdots & & 1 & & \\ & \ddots & & & \cdots & & \ddots & \\ & & 1 & & & \cdots & & 1_u \\ & & & \ddots & & & & \\ & & & & 1_{2M} & & \cdots & \end{bmatrix}_{2M\times L}$$

$$[\Lambda_p]_{L\times L} = \begin{bmatrix} p(-\frac{L-1}{2}) & & & & \\ & \ddots & & & \\ & & p(i - \frac{L-1}{2}) & & \\ & & & \ddots & \\ & & & & p(\frac{L-1}{2}) \end{bmatrix}_{L\times L}$$

We define

$$\Lambda_1 = \begin{bmatrix} e^{-j\pi(-\frac{M-1}{2})(L-1)/M} & & & & \\ & \ddots & & & \\ & & e^{-j\pi(i-\frac{M-1}{2})(L-1)/M} & & \\ & & & \ddots & \\ & & & & e^{-j\pi(\frac{M-1}{2})(L-1)/M} \end{bmatrix}_{M\times M}$$

$$\Lambda_2 = \begin{bmatrix} e^0 & & & & \\ & \ddots & & & \\ & & e^{-j\pi i(M-1)/M} & & \\ & & & \ddots & \\ & & & & e^{-j\pi(M-1)(M-1)/M} \end{bmatrix}_{M\times M}$$

The M point IDFT matrix W_M^* is defined as:

$$W_M^* = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{\frac{j2\pi}{M}} & \cdots & e^{\frac{j2\pi(M-1)}{M}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{\frac{j2\pi(M-1)}{M}} & \cdots & e^{\frac{j2\pi(M-1)(M-1)}{M}} \end{bmatrix}_{M\times M}$$

Since $s_n = H_s^T x_n$ (eq. 5.5), it can be expressed as:

$$\begin{aligned} s_n &= (T \times [I_M \ (-1)^{M-1} I_M] \times [I_{2M} \ I_{2M} ... \ I_{2M} \ \hat{I}_{2M,u}] \times \Lambda_p)^T x_n \\ &= (\Lambda_1 W_M^* \Lambda_2 \times [I_M \ (-1)^{M-1} I_M] \times [I_{2M} \ I_{2M} ... \ I_{2M} \ \hat{I}_{2M,u}] \times \Lambda_p)^T x_n \end{aligned}$$

Since $\Lambda_p^T = \Lambda_p$ and $(\Lambda_1 W_M^* \Lambda_2)^T = \Lambda_2 W_M^* \Lambda_1$, we have:

$$\begin{aligned} s_n = \Lambda_p \times [I_{2M} \ I_{2M} ... \ I_{2M} \ \hat{I}_{2M,u}]^T \times \\ [I_M \ (-1)^{M-1} I_M]^T \times \Lambda_2 W_M^* \Lambda_1 x_n \end{aligned}$$