UNIVERSITAT POLITÈCNICA
DE CATALUNYA

CTTC
Centre Tecnològic
de Telecomunicacions de Catalunya

Digital design and experimental validation of high-performance real-time OFDM systems

Ph.D. Dissertation

Josep Oriol Font Bach

Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)
e-mail: oriol.font@cttc.cat

Ph.D. Thesis Advisors:

Dr. Antonio Pascual Iserte
Associate Professor
Departament de Teoria del Senyal i Comunicacions (TSC)
Universitat Politècnica de Catalunya (UPC)
e-mail: antonio.pascual@upc.edu

Dr. Nikolaos Bartzoudis
Senior Researcher
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)
e-mail: nikolaos.bartzoudis@cttc.cat

Barcelona, July 2013

M'han dit les veus assenyades
que era inútil cansar-me;
però a mi un somni mai no em cansa,
i malgrat la meva barba
sóc infant en la mirada.

Lluís Llach, Vida (1979)

Well I'm sorry but I'm not interested
in gold mines, oil wells, shipping or real estate
What would I liked to have been?
Everything you hate.

Jack White, The Union Forever (2001)

Abstract

The goal of this Ph.D. dissertation is to address a number of challenges encountered in the digital baseband design of modern and future wireless communication systems. The fast and continuous evolution of wireless communications has been driven by the ambitious goal of providing ubiquitous services that could guarantee high throughput, reliability of the communication link and satisfy the increasing demand for efficient re-utilization of the heavily populated wireless spectrum. To cope with these ever-growing performance requirements, researchers around the world have introduced sophisticated broadband physical (PHY)-layer communication schemes able to accommodate higher bandwidth, which indicatively include multiple antennas at the transmitter and receiver and are capable of delivering improved spectral efficiency by applying interference management policies.

The merging of Multiple Input Multiple Output (MIMO) schemes with the Orthogonal Frequency Division Multiplexing (OFDM) offers a flexible signal processing substrate to implement the PHY-layer of various modern wireless communication systems. This is mainly due to the fact that this technology combination is able to provide increased channel capacity and robustness against multipath fading channels. Additionally, Orthogonal Frequency Division Multiple Access (OFDMA) is augmenting the capacities of the MIMO-OFDM technology to serve various mobile subscribers at the same time. A prominent scheme proposed to capitalize the benefits of diversity is the closed-loop MIMO communications, where the receiver is providing information to the transmitter related to the current channel conditions by means of a dedicated feedback channel. In the transmitter, the Channel State Information (CSI) is exploited to adapt at run-time the transmission and, thus, take advantage of the capacities provided by MIMO-OFDM(A).

The increased performance and flexible PHY-layer features of communication systems featuring MIMO-OFDM come at a cost of an increased computational load at baseband. Thus, innovating algorithmic, design and implementation solutions are required to provide the required PHY-layer schemes. Indeed, many levels of innovation are required to pass from a high-level model-based description of the system and its embedded algorithms to their digital realization. In fact, innovating digital design techniques aiming at maximizing the parallelization and resource re-utilization of the baseband Digital Signal Processing (DSP) algorithms have to be employed towards this end, in order to efficiently realize advanced PHY-layer schemes based on the MIMO-OFDM technology.

The Field-Programmable Gate Array (FPGA) custom-processing devices were selected for realizing the proof-of-concept developments of the thesis. When facing the real-time implementation of custom bit-intensive DSP architectures, FPGAs satisfy a trade-off between performance and flexibility, whilst providing the proof-of-concept environment, where innovating Register Transfer Level (RTL)-design techniques can be realistically validated. Moreover, a custom Hardware Description Language (HDL) coding approach was adopted to optimally define the processing demanding RTL designs.

An important aspect of the presented PHY-layer prototyping is the utilization of real-life operating conditions, hardware specifications, constraints and mobile channel propagation conditions. This was made feasible by using the GEDOMIS® testbed. GEDOMIS is a high capacity signal generation and signal processing platform that enables the end-to-end real-time prototyping of multi-antenna wireless communication systems.

In order to support the complex development cycle it was introduced an iterative design, implementation and verification methodology, covering all the required steps from the definition of the system requirements and high-level modelling to the comprehensive evaluation of the resulting prototype based on a realistic hardware-setup.

The core contribution of this thesis is the simplification and optimization of a number of DSP operations that form part of certain baseband building blocks, which are encountered in modern OFDM-based communication systems. This not only allowed to meet the stringent real-time performance requirements, but also enabled the intelligent re-utilization, resource sharing or parallelization of the processing and memory resources available at the target FPGA devices.

Two representative use cases detail this core contribution and underlined the suitability of the proposed development flow. The first use case is a 2x2 MIMO closed-loop PHY-layer scheme, based on the mobile Worldwide Interoperability for Microwave Access (WiMAX) wireless communication standard and featuring a Transmit Antenna Selection (TAS) mechanism. The second use case implements a 3rd Generation Partnership Project (3GPP) Long Term Evolution (LTE)-based macrocell/femtocell interference-management scheme.

Resum

Aquesta tesi doctoral té com a objectiu abordar diversos reptes que es presenten en el disseny digital en banda base per a sistemes de comunicació sense fils moderns i futurs. La ràpida i continua evolució que han experimentat les comunicacions sense fils ha estat promoguda per l'ambiciós objectiu de proveir de serveis omnipresents capaços de garantir un alt rendiment i la fiabilitat dels enllaços de comunicació, tot satisfent el creixent requeriment per a la reutilització eficient de l'atapeït espectre radioelèctric. Per tal de fer front als requeriments de rendiment sempre creixents, investigadors d'arreu del món han proposat sofisticats esquemes de comunicació de capa física (PHYsical-layer, PHY), capaços d'acomodar un elevat ample de banda, incloent també de forma indicativa l'ús de múltiples antenes tant al transmissor com al receptor i millorant l'eficiència espectral mitjançant l'aplicació de principis per a la gestió de la interferència.

La combinació d'esquemes multi-antena (Multiple Input Multiple Output, MIMO) amb la Multiplexació per Divisió Ortogonal de Freqüència (Orthogonal Frequency Division Multiplexing, OFDM) ofereix un medi flexible de processat del senyal per a la implementació de la capa física de diversos sistemes de comunicacions sense fils moderns. Això és degut al seu potencial per a proveir d'una major capacitat de canal i protecció envers els diferents retards i pèrdues introduïdes per les transmissions sobre canals multi-trajectòria. Addicionalment, la Divisió Ortogonal de Freqüència d'Accés Múltiple (Orthogonal Frequency Division Multiple Access, OFDMA) augmenta les capacitats de la tecnologia MIMO-OFDM permetent servir simultàniament a diversos subscriptors mòbils. Un esquema MIMO prominent que permet capitalitzar els beneficis de la diversitat és el de llaç tancat, on el receptor proveeix Informació sobre l'Estat del Canal (Channel State Information, CSI) al transmissor mitjançant un enllaç de retorn dedicat. El transmissor explota la CSI per a adaptar la transmissió dinàmicament i així poder aprofitar completament les capacitats de la tecnologia MIMO-OFDM(A).

L'elevat rendiment i les capacitats flexibles de la capa física dels sistemes de comunicació que utilitzen MIMO-OFDM impliquen un augment de la càrrega computacional en banda base. Conseqüentment, calen solucions innovadores a nivell algorítmic, de disseny i d'implementació per tal de proveir els esquemes de capa física requerits. De fet, es requereix d'innovació en diversos estrats per a passar de la descripció d'alt nivell del sistema i els algorismes que el formen (i.e., modelat) a la seva realització digital. Concretament, cal utilitzar tècniques de disseny digital innovadores que tinguin com a objectiu maximitzar el paral·lelisme i la reutilització de recursos per a la implementació dels algorismes de Processat Digital del Senyal (Digital Signal Processing, DSP), permetent així assolir una realització eficient d'esquemes avançats de capa física basats en la

tecnologia MIMO-OFDM.

Els dispositius Matriu de Portes Programables (Field-Programmable Gate Array, FPGA), caracteritzats per possibilitar un processament a mida, es van escollir per a realitzar els desenvolupaments de prova de concepte de la tesi. Quan es considera la implementació en temps real d'arquitectures a mida d'alt rendiment per al DSP, les FPGAs proveeixen d'alta capacitat computacional i flexibilitat, oferint doncs un entorn ideal per a la validació experimental de tècniques innovadores de disseny a Nivell de Transferència de Registres (Register Transfer Level, RTL). Concretament, s'ha escollit una aproximació basada en la creació de codi a mida usant un Llenguatge de Descripció de Maquinari (Hardware Description Language, HDL), que permet definir de forma òptima aquells dissenys RTL que presenten elevats requeriments de computació.

Un aspecte important del propotipat de capa física que es presenta és la utilització de condicions realistes d'operació, d'especificacions i restriccions de maquinari i de condicions de propagació en canals mòbils. Això va ésser possible gràcies a la utilització del banc de proves GEDOMIS®. GEDOMIS és una plataforma d'alta capacitat per a la generació i processat del senyal, que permet el prototipat complet en temps real de sistemes multi-antena de comunicacions sense fils.

Per tal de suportar el complex cicle de desenvolupament es va introduir una metodologia iterativa de disseny, implementació i verificació, cobrint totes els passos des de la definició dels requeriments i modelat d'alt nivell del sistema fins a la validació exhaustiva dels prototipus resultants mitjançant una plataforma realista.

La contribució central de la tesi és la simplificació i optimització de diverses operacions de DSP que formen part de blocs arquitecturals de la banda base, que es troben als sistemes moderns de comunicació basats en OFDM. Això no només va permetre de satisfer els estrictes requeriments de rendiment impostats per la seva operació en temps real, sinó que ha habilitat la reutilització intel·ligent, compartició o paral·lelització dels recursos de processament i d'emmagatzemament proveïts pels dispositius FPGA escollits per a la seva implementació.

Dos casos d'ús detallen aquesta contribució central, subratllant la conveniència del flux de desenvolupament proposat. El primer cas d'us és un esquema MIMO de llaç tancat, basat en l'estàndard de comunicacions sense fils WiMAX mòbil i emprant un mecanisme de Selecció d'Antena Transmissora (Transmit Antenna Selection, TAS). El segon cas d'ús implementa un esquema de gestió de la interferència en un escenari macro-cel·la/femto-cel·la basat en l'estàndard 3GPP LTE.

Resumen

La presente tesis doctoral tiene como objetivo principal abordar varios retos que se presentan en el diseño digital en banda base para sistemas de comunicación inalámbrica modernos y futuros. La rápida y continua evolución que han experimentado las comunicaciones inalámbricas ha estado promovida por el ambicioso objetivo de proveer de servicios omnipresentes capaces de garantizar un elevado rendimiento y la fiabilidad de los enlaces de comunicación, a la vez que se satisface el creciente requerimiento para la reutilización eficiente del saturado espectro radioeléctrico. Con el fin de hacer frente a los requisitos de rendimiento siempre crecientes, investigadores de alrededor del mundo han propuesto sofisticados esquemas de capa física (PHYsical-layer, PHY), capaces de acomodar un elevado ancho de banda, incluyendo también de forma indicativa el uso de múltiples antenas tanto en el transmisor como en el receptor y mejorando la eficiencia espectral mediante la aplicación de principios para la gestión de la interferencia.

La combinación de esquemas multi-antena (Multiple Input Multiple Output, MIMO) con la Multiplexación por División de Frecuencias Ortogonales (Orthogonal Frequency Division Multiplexing, OFDM) ofrece un medio flexible de procesamiento de señal para la implementación de la capa física de varios sistemas de comunicación inalámbrica modernos. Esto es debido a su potencial para proveer de una mayor capacidad de canal y protección frente los diferentes retardos y pérdidas introducidas por las transmisiones sobre canales multitrayectoria. Adicionalmente, la División de Frecuencias Ortogonales de Acceso Múltiple (Orthogonal Frequency Division Multiple Access, OFDMA) aumenta las capacidades de la tecnología MIMO-OFDM permitiendo servir simultáneamente a varios suscriptores móviles. Un esquema MIMO prominente que permite capitalizar los beneficios de la diversidad es el de lazo cerrado, donde el receptor provee Información sobre el Estado del Canal (Channel State Information, CSI) al transmisor mediante un enlace de retorno dedicado. El transmisor explota la CSI para adaptar la transmisión dinámicamente y así poder aprovechar completamente las capacidades de la tecnología MIMO-OFDM(A).

El elevado rendimiento y capacidades flexibles de la capa física de los sistemas de comunicación que usan MIMO-OFDM implican un aumento de la carga computacional en banda base. Consecuentemente, se necesitan soluciones innovadoras a nivel algorítmico, de diseño e implementación para poder proveer los esquemas de capa física requeridos. De hecho, se requiere de innovación en varios estratos para pasar de la descripción de alto nivel del sistema y los algoritmos que lo forman (i.e., modelado) a su realización digital. Concretamente, es necesario utilizar técnicas de diseño digital innovadoras que tengan como objetivo maximizar el paralelismo y la reutilización de recursos para la imple-

mentación de los algoritmos de Procesamiento Digital de Señal (Digital Signal Processing, DSP), permitiendo así obtener una realización eficiente de esquemas avanzados de capa física basados en la tecnología MIMO-OFDM.

Los dispositivos Matriz de Puertas Programables (Field-Programmable Gate Array, FPGA), caracterizados por posibilitar un procesamiento a medida, se escogieron para realizar los desarrollos de prueba de concepto de la tesis. Cuando se considera la implementación en tiempo real de arquitecturas a medida de alto rendimiento para el DSP, las FPGAs proveen de alta capacidad computacional y flexibilidad, ofreciendo así un entorno ideal para la validación experimental de técnicas innovadoras de diseño a Nivel de Transferencia de Registros (Register Transfer Level, RTL). Concretamente, se ha seleccionado una aproximación basada en la creación de código a medida usando un Lenguaje de Descripción de Hardware (Hardware Description Language, HDL), que permite definir de forma óptima aquellos diseños RTL que presentan elevados requisitos de computación.

Un aspecto importante del prototipado de capa física que se presenta es la utilización de condiciones realistas de operación, de especificaciones y restricciones de hardware y de condiciones de propagación en canales móviles. Esto fue posible gracias a la utilización del banco de pruebas GEDOMIS®. GEDOMIS es una plataforma de alta capacidad para la generación y procesamiento de señal, que permite el prototipado completo en tiempo real de sistemas multi-antena de comunicación inalámbrica.

Para soportar el complejo ciclo de desarrollo se introdujo una metodología iterativa de diseño, implementación y verificación, cubriendo todos los pasos desde la definición de los requerimientos y modelado de alto nivel del sistema hasta la validación exhaustiva de los prototipos resultantes mediante una plataforma realista.

La contribución central de la tesis es la simplificación y optimización de varias operaciones de DSP que forman parte de bloques arquitecturales de la banda base, que se encuentran en los sistemas modernos de comunicación basados en OFDM. Esto no solo permitió satisfacer los estrictos requisitos de rendimiento impuestos por su operación en tiempo real, sino que ha posibilitado la reutilización inteligente, compartición o paralelización de los recursos de procesamiento y almacenamiento provistos por los dispositivos FPGA escogidos para su implementación.

Dos casos de uso detallan esta contribución central, subrayando la conveniencia del flujo de desarrollo propuesto. El primer caso de uso es un esquema MIMO de lazo cerrado, basado en el estándar de comunicaciones inalámbricas WiMAX móvil y empleando un mecanismo de Selección de Antena Transmisora (Transmit Antenna Selection, TAS). El segundo caso de uso implementa un esquema de gestión de la interferencia en un escenario macro-celda/femto-celda basado en el estándar 3GPP LTE.

Acknowledgements

This is probably the most complicated text to be produced in a thesis. First, it is the only portion you are sure that everyone will read (experts in the field or not). Moreover, you will be judged by what you say or rather for what you don't say. Also, is the only portion of text where only few etiquette rules apply (as you will see I am mixing different languages and non-formal expressions). But more importantly, it is where you are openly revealing your inner thoughts, feelings and gratitude towards relevant people. Let the feast begin!

Primero de todo, quiero agradecer la gran oportunidad que me han dado mis dos co-directores. No solo me habéis obsequiado con vuestro conocimiento e infinita paciencia, sino que me habéis hecho de padres en este fascinante mundo que es la investigación. Junto a vosotros he crecido personal y profesionalmente. Por si fuera poco, me he divertido mucho a vuestro lado (ya sea en Berlín, en Madrid, en Barcelona o en Castelldefels) y sé que nuestros caminos se cruzarán muchas veces más en el futuro. Toni, tu capacidad de aprender cosas nuevas, tu pasión por hacerlo y tus muchas habilidades como profesor nunca dejaran de maravillarme. Además, tu alegría se contagia y no puede ser mejor compañera de trabajo. Νίκο, ϑαυμάζω την ικανότητα ηγεσίας και τη σαφήνειά με την οποία βλέπεις τα πράγματα (όχι μόνο σε επαγγελματικό επίπεδο). Είσαι ένα πρόσωπο που εμπλουτίζει όσους γνωρίζει. Σε σένα δεν έχω βρει μόνο κάποιο για να κοιτάζω προς τα εμπρός, αλλά ένα αληϑινό φίλο. Σε κάϑε περίπτωση να ϑυμάσαι ότι έχω το βίντεο με το ζεϊμπέκικό σου.

Vull agrair al CTTC les moltes facilitats que m'ha ofert: no només he tingut accés al millor equipament que un enginyer pugui desitjar, sinó que m'ha proveït d'un entorn de treball immillorable – càlid i estimulant, tot rodejat d'excel·lència científica i tècnica. Quiero agradecer especialmente a Ana Pérez y a Miguel Ángel Lagunas por introducirme en el desarrollo de sistemas MIMO y por el empuje y ánimos que me han ofrecido en todo momento. També vull agrair particularment l'imprescindible ajuda tècnica, professional i personal que m'ha donat el Miquel Payaró aquests darrers anys.

També m'agradaria donar les gràcies al José López i al Javier Arribas pels esforços de revisió; amb el darrer he tingut la sort de compartir bones estones a un recent viatge a Sophia-Antipolis (la seva elevada experiència i entusiasme per la recerca aplicada han donat peu a bons consells per acabar de retocar el text).

Aquesta tesi s'ha enriquit notablement gràcies a les aportacions dels companys amb qui he tingut la sort de treballar al CTTC. Vull començar destacant al David López, amb qui he pogut comptar en tot moment (podríem dir que tot va començar a Budapest...); per si això fos poc, la seva ajuda ha resultat fonamental a les llargues hores que hem passat al laboratori. Voglio ringraziare soprattutto l'amicizia e il supporto (su molti fronti) che Sor Paolo Dini mi ha offerto. Alla fine non siamo riusciti a costruire un quadro comune per la ricerca, ma (indipendentemente da ciò che accadrà in futuro) quello che abbiamo guadagnato è molto più importante, caro amico. Cal també una menció especial al Xavier Artiga, Marco Miozzo, Lluís Parcerisa i David Pubill pels molts àpats i litres de cafè que hem pres junts mentre canviàvem el món. Evidentment, també vull agrair especialment l'ajuda a la resta d'equip de BeFEMTO-EngU: Luís Blanco, Marc Molina, Pepe Rubio i Jordi Serra. Igualment, vull mencionar l'ajuda del Lluís Ventura durant els meus primers mesos al CTTC. Al José Núñez li he d'agrair una llarga col·lecció de pel·lícules de sèrie B que m'han ajudat a evadir-me quan semblava que la tesi podia amb mi. Per acabar, vull donar les gràcies al Josep Mangues per l'ajuda i consells rebuts, ja sigui durant la meva petita incursió a la tercera capa o posteriorment.

Com no podria ser d'una altra manera, la part més personal d'aquest text havia de començar amb tu; Anna, crec que molt poca gent pot dir que ha tingut la sort de trobar a la persona que el complementa, aquella ànima a qui un sembla ser predestinat (allò que en diuen l'amor de la seva vida). Ets la meva millor amiga, la meva companya i qui em dóna la força per a voler superar-me, per a evitar caure en el conformisme i la immobilitat que comporten la quotidianitat de la vida adulta. Sé que el nostre vincle és molt especial i que es fa més fort cada dia, creixent juntament amb el nostre fill. Sento que junts tot ho podem. Com diu el Jules "two can be complete without the rest of the world".

I arribem al més petit de casa, però també el més important de tots. Jac, encara que se que trigaràs molt de temps a comprendre realment el significat d'aquestes quatre ratlles, vull dir-te que la meva felicitat va néixer amb el primer batec del teu cor. Veure't créixer i aprendre de forma tant vertiginosa, però agafat de la meva mà, és fascinant i, sens dubte, el què dóna sentit al meu dia a dia. No deixis mai que et convencin de que hi ha res que no pots aconseguir. I no abaixis el cap davant de ningú, sé que arribaràs allà on potser jo no he sabut, petit Indy. T'estimo més del que mai hauria cregut de ser capaç!

Pares, he tingut la sort de créixer al vostre costat. No només m'heu guiat, animat, ajudat i deixat volar quan així ho he volgut, sinó que m'heu estimat incondicionalment, descobert el món i, encara ara, protegit sota la vostra ala. Sé que aquesta tesi us farà més contents, si cap, a vosaltres que a mi; jo estaré content si mai arribo a acostar-me al què vosaltres representeu per a mi. I tu, cap de trons, ja hauries de saber com d'important ets per a mi. Tens moltes més qualitats, habilitats i força de la que ets capaç de veure; sinó perquè et penses que el teu germà gran sempre ha volgut assemblar-se a tu? Escriu poeta, pensa filòsof, però sobretot no dubtis de que meravellaràs al món, Madrona. Avis, malgrat ja no sigueu aquí, vull dir-vos heu sigut una part fonamental de la meva vida. Espero que les vostres creences siguin certes i que, en algun lloc incorpori, us hagueu retrobat i es tornin a sentir les vostres afectuoses renyines.

Vull acabar donant les gràcies a la resta de la meva (per sort) gran família. Tenías razón *Totoy*, eres un buen tío. Montse gràcies per fer-me sentir com un fill més a casa teva. A l'Eva i a l'Iñaki (més ben dit, als pares del Roc) els he d'agrair especialment les moltes hores i afecte que han dedicat al Jac (fins us perdono el tema de l'espanyol!).

Contents

Acronyms **xv**

1 Introduction **1**
1.1 Thesis overview . . . 8

2 The PHY-layer development ecosystem **9**
2.1 Introduction to the PHY-layer development ecosystem . . . 9
2.1.1 Fundamental research . . . 10
2.1.2 Realistic PHY-layer modelling . . . 14
2.1.3 PHY-layer implementation and prototyping . . . 19
2.1.4 Large-scale IC production . . . 30
2.1.5 Synopsis . . . 32
2.2 Motivation . . . 34

3 Related work and contribution **37**
3.1 Preamble . . . 37
3.1.1 Definition of basic nomenclature . . . 37
3.1.2 Research groups used as a primary reference . . . 39
3.2 Review of related work . . . 41
3.2.1 Advanced PHY-layer modelling initiatives . . . 41
3.2.2 PHY-layer implementation and prototyping efforts . . . 45
3.2.3 Efficient design targeting IC-technology . . . 55
3.3 Objectives, assumptions and conditions for the developments . . 61
3.4 Contribution . . . 67

4 Design, implementation and verification methodology **73**
4.1 Considered development framework . . . 73
4.1.1 Considered development platform . . . 74
4.2 Proposed design, implementation and validation methodology . . 76
4.2.1 Multi-stage testing strategy . . . 76
4.2.2 Incremental development flow . . . 77
4.2.3 Benefits of the proposed methodology . . . 84
4.3 Development-flow challenges . . . 85
4.4 Description of the GEDOMIS® testbed . . . 88
4.4.1 RF front-ends . . . 89
4.4.2 Channel emulation . . . 90
4.4.3 Baseband processing elements . . . 92
4.4.4 Complimentary hardware components . . . 94

4.4.5 GEDOMIS' signal impairments 96
4.4.6 Utilization of GEDOMIS 98

5 Use Case I 103
5.1 Considered system 103
5.1.1 Basic description of the MIMO-OFDM technology 103
5.1.2 Short introduction to the Mobile WiMAX PHY-layer . . 105
5.1.3 System specifications and included features 109
5.2 Utilizing an incremental development 109
5.2.1 Single-antenna open-loop scheme 110
5.2.2 1x2 SIMO open-loop scheme 117
5.2.3 2x2 MIMO STBC-based system 118
5.2.4 2x2 MIMO closed-loop scheme 122
5.3 Innovating RTL design techniques 129
5.3.1 Synchronization 129
5.3.2 Channel estimation architecture 132
5.3.3 Joint design of the STBC and MRC techniques 135
5.3.4 Adaptive memory structure 139
5.3.5 Implementing operations related to the WiMAX standard 150
5.3.6 Centralized control unit 154
5.4 Integration and implementation using the GEDOMIS testbed . . 156
5.4.1 Real-time MIMO signal transmission 157
5.4.2 Real-time MIMO channel emulation 160
5.4.3 Real-time MIMO signal reception 160
5.4.4 FPGA-based implementation of the presented systems . . 164
5.5 Experimental results 166

6 Use Case II 171
6.1 Considered system 171
6.1.1 Basic introduction to opportunistic frequency reuse 171
6.1.2 Short introduction to the 3GPP LTE PHY-layer 172
6.1.3 Considered scenario and PHY-layer specifications 174
6.1.4 Interference management scheme 177
6.2 Utilizing an incremental development 178
6.2.1 Operation of the extended DFE 180
6.3 Efficient RTL design 184
6.3.1 Hardware-efficient digital filtering stage 184
6.3.2 Synchronization and interference-detection 186
6.3.3 Centralized control unit 188
6.4 Integration and implementation using the GEDOMIS testbed . . 189
6.4.1 Real-time channel and interference emulation 189
6.4.2 FPGA-based implementation 190
6.5 Experimental results 191

7 Conclusions and future work lines 197
7.1 Conclusions 197
7.2 Future work lines 200

Bibliography 201

Acronyms

3GPP Third Generation Partnership Project

4G Fourth Generation

ADC Analog-to-Digital Converter

ADP Advanced Development Platform

AGC Automatic Gain Control

AI Applied Instruments

ALU Arithmetic Logical Unit

AMC Adaptive Modulation and Coding

API Application Programming Interface

ASIC Application Specific Integrated Circuit

AWGN Additive WGN

BER Bit Error Rate

BICM Bit Interleaved Coded Modulation

BS Base Station

BW BandWidth

BWA Broadband Wireless Access

CAMTA Argentine Conference on Micro-Nanoelectronics, Technology, and Applications

CBEA Cell Broadband Engine Architecture

CCI Co-Channel Interference

CFO Carrier Frequency Offset

CIR Carrier-to-Interference Ratio

CMOS Complementary Metal-Oxide-Semiconductor

CoMP Coordinated Multi-Point

CORDIC Coordinate Rotation Digital Computer

COTS Commercial-Off-The-Shelf

CP Cyclic Prefix

cPCI compact PCI

CPE Customer Premises Equipment

CR Cognitive Radio

CS Compressed-Sensing

CSI Channel State Information

CTTC Centre Tecnològic de Telecomunicacions de Catalunya

EB ElektroBit

DAC Digital-to-Analog Converter

DC Direct Current

DDC Digital Down Converter

DDS Direct Digital Synthesizer

DFE Digital Front-End

DL DownLink

DSP Digital Signal Processing

EDA Electronic Design Automation

EUSIPCO European Signal Processing Conference

ETH Eidgenössische Technische Hochschule

EVM Error Vector Magnitude

FCH Frame Control Header

FDATool Filter Design and Analysis Tool

FDD Frequency Division Multiplexing

FEC Forward Error Correction

FIFO First In First Out

FIR Finite Impulse Response

FFT Fast Fourier Transform

FPGA Field Programmable Gate Array

FTW Forschungszentrum Telekommunikation Wien

FUSC Full Usage of the Subchannels

GEDOMIS GEneric hardware DemOnstrator for MIMO Systems

GPGPU General-Purpose (computation on) GPU

GPP General Purpose Processor

GPU Graphics-Processing Unit

GTAS Grupo de Tratamiento Avanzado de Señal (UC research group)

GTEC Grupo de Tecnología Electrónica y Comunicaciones (UDC research group)

GUI Graphical User Interface

HDL Hardware Description Language

HetNet Heterogeneous Networks

HHI Heinrich-Hertz-Institut

HIL Hardware-In-the-Loop

HLPL High-Level Programming Language

HLS High-Level Synthesis

HSDPA High-Speed DL Packet Access

HW/SW HardWare/SoftWare

GBPS Giga-Byte Per Second

I/O Input/Output

I/Q In-phase and Quadrature

IC Integrated Circuit

ICI Inter-Cell Interference

ICIC ICI Coordination

ICT Information and Communication Technologies

IDE Integrated Development Environment

IEEE Institute of Electrical and Electronics Engineers

IF Intermediate Frequency

IFFT Inverse FFT

IMEC Interuniversitair Micro-Elektronica Centrum

IMT International Mobile Telecommunications

IP Intellectual Property

ISI Inter-Symbol Interference

ITU International Telecommunications Union

ITU-R ITU - Radio Communication Sector

KPI Key Performance Indicator

LAN Local Area Network

LDPC Low-Density Parity Check

LNA Low-Noise Amplifier

LNICST Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering

LO Local Oscillator

LST Layered Space-Time

LT Linear Technology

LTE Long Term Evolution

LUT Look-Up-Table

MAC Medium Access Control

MASCOT Multiple-Access Space-Time Coding Testbed

MCS Mercury Computer Systems

MG Major Group

MIMO Multiple Input Multiple Output

MISO Multiple Input Single Output

MLD Maximum-Likelihood Detector

MOBILIGHT Mobile Lightweight Wireless Systems

MPI Message Passing Interface

MRC Maximum Ratio Combining

MSPS Mega-Samples Per Second

MU Multi-User

MUBF MU BeamForming

NCO Numerically Controlled Oscillator

NRE Non Recurring Engineering

OFDM Orthogonal Frequency Division Multiplexing

OFDMA Orthogonal Frequency Division Multiple Access

OS Operating System

PA Power Amplifier

PAPR Peak-to-Average Power Ratio

PAR Place And Route

PCI Peripheral Component Interconnect

PCIe PCI express

PGA Programmable Gain Amplifier

PHY PHYsical

PRB Physical Resource Block

PRBS PseudoRandom Binary Sequence

PSS Primary Synchronization Signal

PUSC Partial Usage of Subchannels

QAM Quadrature Amplitude Modulation

QoS Quality of Service

QPSK Quadrature Phase-Shift Keying

RAM Random Access Memory

RF Radio Frequency

R&S Rohde & Schwarz

RS Reference Signal

RTL Register Transfer Level

SDR Software Defined Radio

SDRAM Synchronous Dynamic RAM

SFDM Space-Frequency Block Code

SIC Successive Interference Cancellation

SINR Signal-to-Interference-plus-Noise Ratio

SIMO Single Input Multiple Output

SIR Signal-to-Interference Ratio

SISO Single Input Single Output

SM Spatial Multiplexing

SMQUAD Signal Master Quad

SNR Signal-to-Noise Ratio

SoC System-On-a-Chip

SSS Secondary Synchronization Signal

STBC Space-Time Block Coding

STBD Space Time Block Decoding

STC Space-Time Coding

STTC Space-Time Trellis-Coding

SVD Singular Value Decomposition

TAS Transmit Antenna Selection

TDD Time Division Duplex

TI Texas Instruments

TridentCom Testbeds and Research Infrastructure for the Development of Networks and Communities

TSC Teoria del Senyal i Comunicacions (UPC department)

TU Technischen Universität

UL UpLink

UC Universidad de Cantabria

UDC Universidade da Coruña

UE User Equipment

UPC Universitat Politècnica de Catalunya

VHSIC Very High Speed Integrated Circuit

VHDL VHSIC HDL

VLSI Very-Large-Scale Integration

VSA Vector Signal Analyser

VSG Vector Signal Generator

WARP Wireless Open-Access Research Platform

WGN White Gaussian Noise

WiMAX Worldwide Interoperability for Microwave Access

WiMob Wireless and Mobile Computing, Networking and Communications

WLAN Wireless LAN

WSN Wireless Sensor Network

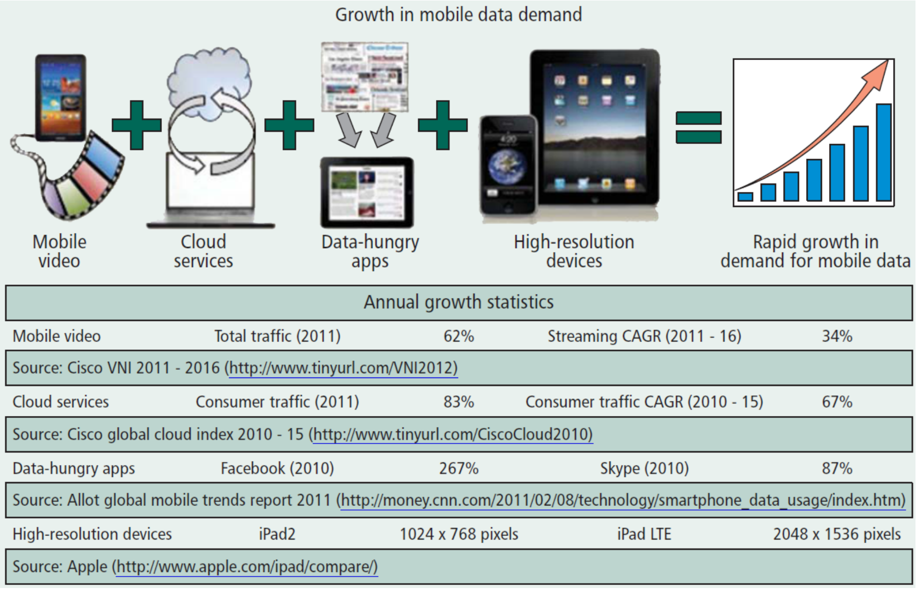
Chapter 1

Introduction

In recent years the world has experienced a dramatic evolution of wireless communications. This fast and continuous progression has been driven by the ambitious goal of providing ubiquitous communications services in every time more demanding operating conditions. A continuous increase of throughput and reliability, both for voice and data services, is expected in an environment featuring high mobility and a network composed by heterogeneous technologies. Novel solutions have been proposed by the academic community and the industry towards this end. In fact, the academic community has been covering a wide range of objectives from fundamental to applied research. Furthermore, the collaboration between both actors has been leading the generation of medium-to-long term solutions, aiming at setting the basis of future communication technologies. The target of the present thesis is the construction of solutions that will be employed in real-life applications in the mid-term.

Mobile technology has entirely reshaped the telecommunications landscape, increasing the number of subscribers exponentially during the past decade. From the infrastructure point of view, mobile networks have suffered a drastic evolution in response to the ever increasing user requirements. Mobile operators have grown to dominate the market, offering services as comprehensive as those offered by their wireline counterparts. Furthermore, the utilization that the end-users are making of mobile phones and other mobile devices has also seen a significant change. The advanced mobile phones (e.g., smartphones) and other mobile devices have moved way beyond voice services, enabling complex data-hungry applications (e.g., multimedia, video-calls, social networking, interactive multi-user applications) which have led to an unprecedented demand for high performance, high capacity and high bit-rate (see Figure 1.1).

A key player in the future of mobile communications are the Broadband Wireless Access (BWA) technologies, which are meant to satisfy the advanced high data-rate and mobility requirements of the so-called 4G (fourth generation) and beyond mobile communications systems. The 4G requirements are detailed in the International Mobile Telecommunications Advanced (IMT-Advanced) specifications [ITU-R, 2003] defined by the International Telecommunications Union - Radio Communications Sector (ITU-R). The baseline data-rate for a 4G service is set at 100 Mbit/s for high mobility communication (i.e., speeds up to 500 km/h) and 1 Gbit/s for low mobility (i.e., up to 10 km/h) and stationary communication, as illustrated in Figure 1.2. Two indicative standard-


Growth in mobile data demand
Mobile video
Cloud services
Data-hungry apps
High-resolution devices
Rapid growth in demand for mobile data

Annual growth statistics				
Mobile video	Total traffic (2011)	62%	Streaming CAGR (2011 - 16)	34%
Source: Cisco VNI 2011 - 2016 (http://www.tinyurl.com/VNI2012)				
Cloud services	Consumer traffic (2011)	83%	Consumer traffic CAGR (2010 - 15)	67%
Source: Cisco global cloud index 2010 - 15 (http://www.tinyurl.com/CiscoCloud2010)				
Data-hungry apps	Facebook (2010)	267%	Skype (2010)	87%
Source: Allot global mobile trends report 2011 (http://money.cnn.com/2011/02/08/technology/smartphone_data_usage/index.htm)				
High-resolution devices	iPad2	1024 x 768 pixels	iPad LTE	2048 x 1536 pixels
Source: Apple (http://www.apple.com/ipad/compare/)				

Figure 1.1: Relevant summary statistics on the annual growth of factors contributing to data explosion [1].

ization initiatives towards 4G wireless communication are the 3rd Generation Partnership Project (3GPP) Long Term Evolution (LTE), with its latest revision LTE-Advanced, and the Worldwide Interoperability for Microwave Access (WiMAX), through its evolved WiMAX-2 specifications (Institute of Electrical and Electronics Engineers - IEEE - 802.16.m) [2]. In spite of the ambitious requirements defined for 4G systems, future applications will continue impulsing the evolution of both mobile devices and service infrastructures, imposing demands that nowadays are only envisioned in a fundamental research context.

In addition, the different requirements imposed by applications which are developed in a constantly evolving wireless ecosystem, is promoting the coexistence of diverse wireless access technologies. This has given birth to the concept of Heterogeneous Networks (HetNet) posing serious design problems both at network and terminal level (e.g., the modern mobile devices are required to support multiple wireless access technologies).

In an over-crowded Radio Frequency (RF) spectrum (see Figure 1.3), a determinant factor to cope with the continuously increasing demands of emerging wireless services is that of high spectral efficiency (i.e., high aggregated cell data rate per unit of spectrum). Among other measures, more efficient signal processing techniques, an improved Medium Access Control (MAC) and novel deployment concepts are needed for the the mobile cellular networks. For instance, an increased Base Station (BS) deployment combined with frequency reuse techniques should be implemented in HetNet scenarios. This actually constitutes a major topic investigated under the umbrella of Cognitive Radio (CR). Cognitive devices should be able to opportunistically utilize those RF spectrum bands which are not utilized by the primary communication (i.e., with high priority), without affecting their perceived service quality. A novel deployment scheme is that of small low-power cells coexisting with the evolved

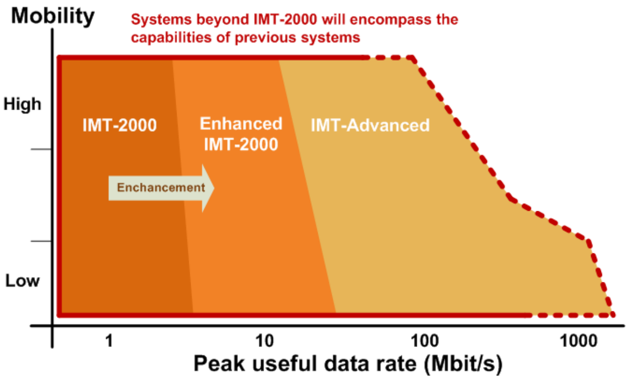


Figure 1.2: Capabilites of IMT-Advanced [ITU-R, 2003].

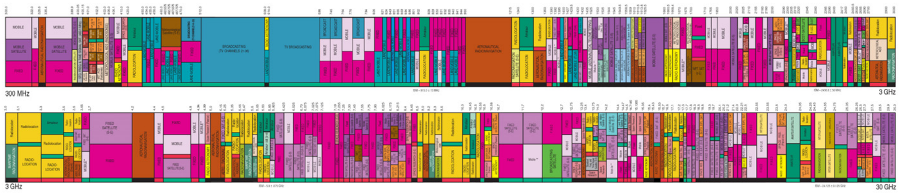


Figure 1.3: United States radio frequency spectrum allocation (300 MHz to 30 GHz band) [NTIA, 2011].

cellular network. Such schemes are denoted as Femtocells (or jointly named as small cells, which encompass a more diverse size of cells) and are mainly aimed at covering the needs of short-range residential gateways, both addressing the indoor topology looses and opportunistically reusing the RF spectrum. However, in order to ensure a ubiquitous end-user experience, it is of paramount importance to guarantee a minimum Quality of Service (QoS).

In order to confront with the challenges encountered in this vast evolution, innovating algorithmic, design and implementation solutions are required at each communication layer (i.e., from network infrastructure to digital baseband design) separately, jointly and/or at system-level. Even if we constrain our view to the digital baseband design (which is the main focus of this thesis), there are still many levels of innovation required to pass from a high-level model-based description of the system and its embedded algorithms to their digital realization. In practical research terms, implementing the physical (PHY)-layer of modern mobile devices, including very complex signal processing, is of paramount importance to enable the real-world deployment of advanced wireless communication systems.

The PHY-layer of next generation wireless communication systems is due to support significantly augmented bandwidths (which implies a massive computational load at baseband), an increased number of users per cell and other flexible specifications that mostly affect the design and overall architecture of the baseband communications. Additionally, it is expected that the utilization of multiple antennas at both the transmitter and receiver sides, known as Multiple Input Multiple Output (MIMO), and interference mitigation techniques will have to be widely adopted to cope with the performance and reliability requirements of future mobile devices.

MIMO communication systems exploit spatial diversity to provide unprecedented capacity and throughput. On the other hand, Orthogonal Frequency Division Multiplexing (OFDM), with several data streams being transmitted in parallel, provides resilience against interference caused by multipath propagation. The merging of MIMO and OFDM offers a flexible signal processing substrate to implement the PHY-layer of various modern wireless communication systems. This is mainly due to the fact that this technology combination is able to provide increased channel capacity and robustness against multipath fading channels. Hence, many recent standardization efforts are using MIMO-OFDM as the basis of the proposed technologies (e.g., Wireless Local Area Network - WLAN, IEEE 802.11a/n/ac, LTE, WiMAX). Undoubtedly, accommodating high-performance prerequisites under fast channel fading implies challenging signal processing at baseband. The latter is scaled when considering a scenario with multiple competing users, where each user expects a minimum service quality (e.g., data-rate, Bit Error Rate - BER). Orthogonal Frequency Division Multiple Access (OFDMA) is augmenting the capacities of the MIMO-OFDM technology to serve various mobile subscribers at the same time. Moreover, a prominent scheme proposed to capitalize the benefits of diversity is the closed-loop MIMO communications, where the receiver is providing information to the transmitter related to the current channel conditions by means of a dedicated feedback channel. In the transmitter, the Channel State Information (CSI) is exploited to adapt at run-time the transmission and, thus, take advantage of the capacities provided by MIMO-OFDM(A).

Another parameter of modern communication systems that every time acquires a more critical role, reducing operating costs [3] or the overall carbon footprint, is provided by optimizing the power consumption, which can also be achieved by means of adaptivity (e.g., accounting for the energetic cost and current channel conditions to select the most adequate MIMO scheme at every moment). Achieving both high performance and adaptivity augments in orders of magnitude the design complexity of MIMO-OFDM based systems. The most significant aspects in this sense are the number of antennas, the bandwidth and the implemented MIMO scheme. An indicative figure is given by the fact that the processing effort scales linearly with the bandwidth [4]. Currently, large bandwidths up to 20 MHz are considered in the pre-4G versions of both LTE and WiMAX (IEEE 802.16e standard, also denoted as mobile WiMAX). In actual or imminent wireless communications technology terms, bandwidths of up to 100 MHz are considered in order to enable true 4G LTE-Advanced systems.

Achieving the desirable performance in the PHY-layer requires to account for issues originated at a higher layer or at system-level. For instance, a strong limitation existing in dense BS deployments is posed by the presence of Inter-Cell Interference (ICI). Such in-band interference imposes significant constraints to the achievable performance. For this very reason, interference management techniques are considered a de-facto operational prerequisite for the deployment of 4G wireless access networks. Hence, interference management is being strongly considered in the current wireless communication standards, adapting and encapsulating novel PHY-layer techniques. While the leading wireless standards based on the MIMO-OFDM technology (e.g., LTE, WiMAX) can achieve improved spectral efficiency within one cell, ICI is still preventing them from

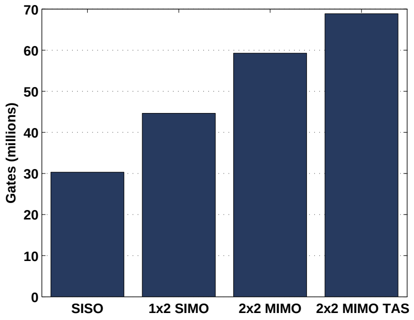


Figure 1.4: Augment of complexity, measured as equivalent gate-counts (Virtex-4 FPGA family), for different transceiver configurations based on results presented in Chapter 5.

coming close to the theoretical rates in multi-cell networks. For the particular case of small cells, many ICI-related innovative solutions are being proposed in the literature to efficiently enable the deployment of LTE-based Femtocells. Adaptive transmission plays a key role in the deployment of frequency reuse and interference management.

The projected exponential growth of wireless communication capacity (i.e., requiring every time for more signal bandwidth and spectral efficiency) caters for ground-breaking innovation not only at algorithmic level, but also new physical Integrated Circuit (IC) technologies, new implementation techniques and design methodologies. This is getting even more complex considering that high performance has to be every time more coupled with energy efficiency (i.e., the technology tendencies are to calculate the processing cost in terms of their bits per joule overhead). Fully exploiting the benefits of the previously introduced PHY-layer innovations, especially when they are combined, relies on the implementation of high-throughput receivers featuring a very high processing complexity at baseband (see Figure 1.4). The latter constitutes a fundamental digital design problem [Zhang, 2001]. As a consequence, the potential of the MIMO-OFDM based technology is still far from being fully exploited in actual implementations. Hence, the efficient implementation of advanced PHY-layer solutions has emerged as a major research topic [Burg, 2006,Edman, 2006, Eberle, 2006,Nilsson, 2007,Camera, 2008,Eberli, 2009,Wenk, 2010].

Addressing the bit-intensive requirements of high performance communication systems is traditionally questing the signal processing community throughout the historical trajectory of digital PHY-layer implementations. On top of that, the constant increase of computational complexity posed by the introduction of bit-intensive PHY-layer solutions is driving the need for massive and parallel computation. A clear trend towards the increased parallelism and heterogeneity is reflected in architectures where the processing elements are utilized to implement demanding Digital Signal Processing (DSP) arithmetic operations and transformations (i.e., PHY-layer of high-performance wireless communication systems), as shown in Figure 1.5. The amount of available options has, however, greatly complicated the selection of the most adequate technology for the digital realization of novel PHY-layer techniques. Mapping the PHY-layer

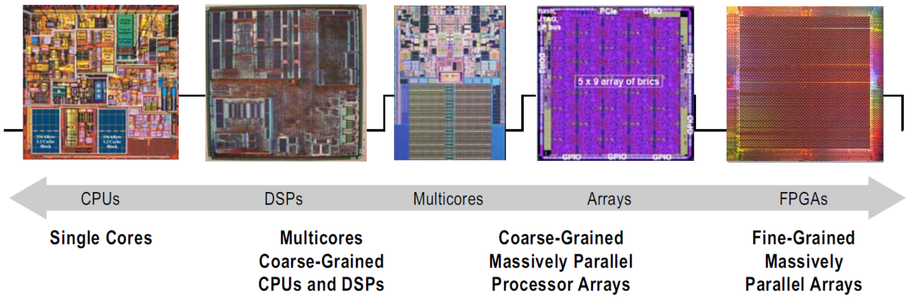


Figure 1.5: Recent trend of programmable and parallel technologies [5].

of different systems to this enormous design-space results in a widely varying performance and energy consumption, not solely when examining the target implementation technology, but also the different application domains and use cases.

Likewise, the selected technology will have a great impact on the costs and time required to implement and validate the system. Hence, innovation is also required in the provision of optimized digital design, in terms of performance, utilized resources and energy consumption, accounting for the limitations of the selected technology. Over the last years, the most prominent technologies used in PHY-layer prototyping have been the DSP processors including multiple cores (usually called multi-cores), the cell processors, the Graphics-Processing Units (GPUs), the Field-Programmable Gate Arrays (FPGAs) and the Application Specific Integrated Circuits (ASICs). Given the large amount of available implementation technologies, the definition of the target application and a (set of) use case(s), results essential to narrow the explorable design-space. Traditionally, computer-based software simulations have constituted an extraordinary tool for the rapid validation and performance assessment of novel PHY-layer algorithms and systems. Nonetheless, the limitations encountered (i.e., assumptions and simplifications are applied considering that the simulation environments feature limited capacity to reproduce dynamic behaviours), seem to indicate that computer-based simulations are just constituting the first step towards the long path required to provide a comprehensive evaluation of novel algorithms and systems for real-life applications. Additionally, as the complexity of the algorithms introduced in the PHY-layer increases, it is not only required additional computational capacity, but also a complex testing infrastructure to accompany the analysis closer to real-world scenarios (e.g., realistic channel conditions, real-time operation, constraints and impairments introduced by the final implementation technology). Hence, the development of prototypes has been a major vehicle to realistically validate and assess the performance of advanced wireless communication systems, diminishing the gap between fundamental research and real-world solutions [Haustein, 2006, Perels, 2007, Caban, 2009, Naya, 2010, Murphy, 2010, Duarte, 2012]. In such context, a robust yet flexible methodology is essential not only to minimize the required design and implementation time, but especially to enable an optimum digital realization and guarantee the quality of the system evaluation in close-to-real-world conditions.

FPGAs are processing devices renown for their capacity to host custom

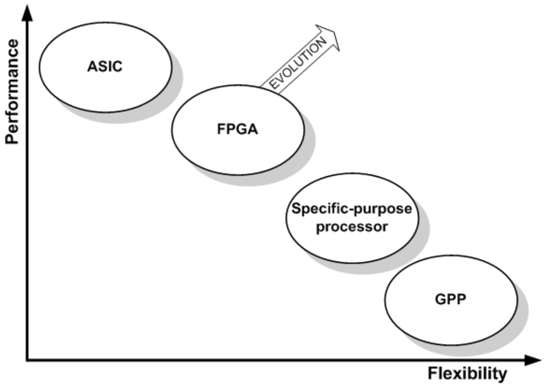


Figure 1.6: Performance vs. flexibility of different PHY-layer implementation technologies [8].

parallel digital processing systems and, thus, able to implement complex DSP-intensive algorithms and advanced wireless communication systems. Modern FPGA devices have gone even further by integrating a number of different dedicated hard-wired Intellectual Property (IP) processing blocks within the fabric (Ethernet, Peripheral Component Interconnect express - PCIe - and Gigabit Input/Output - I/O - cores among others), while increasing dramatically the digital logic and DSP-specific building blocks and the embedded memory capacity. This gives them unprecedented parallel computing capacity from one side and flexibility from the other side (see Figure 1.6), when utilizing run-time partial reconfiguration techniques [6], or when using the on-chip hard-wired multi-core microprocessors [7], available in certain families, following by this way a hardware/software (HW/SW) co-design approach. Therefore, FPGA-based implementations are ideal to meet the needs posed by the development of realistic prototypes (i.e., cost, time, capacity and flexibility). As an example, when prototyping BWA systems, FPGAs are not only appropriate to satisfy the demanding PHY-layer processing requirements, but their flexibility is also very important if the constant evolution of the wireless communication standards is taken into account. Hence, modifications on the standards can be implemented with clearly smoother implications than when compared, for instance, with ASIC-based solutions (which are the preferred solution for large-scale fabrication).

This thesis is focusing on the PHY-layer and more concretely on the efficient realization of baseband signal processing algorithms, by employing advanced digital design techniques and by applying suitable optimizations at digital circuit level. The ultimate goal is to achieve performance improvements with agile usage of baseband processing resources. Such efficient design and FPGA-based implementation of advanced PHY-layer solutions forms the basis of those high-performance wireless communication systems that feature adaptive operation, which follows dynamically variable conditions. FPGAs have been selected because they satisfy a trade-off between performance and flexibility and provide the proof-of-concept environment where innovating Register Transfer Level (RTL)-design techniques can be validated realistically. Moreover, it is very important to underline that the innovative PHY-layer prototyping introduced herein is bound to real-life conditions, hardware specifications and constraints under realistic mobile channel propagation conditions.

The main contribution of this thesis, as it will be elaborated across the following chapters, is the provision of innovation in performance-efficient digital design, which is strongly tight to detailed design principles, procedures and a proposed incremental methodology, that jointly aim at efficiently designing, implementing and realistically verify advanced PHY-layer solutions. Moreover, although the scope of this thesis has been set on high-performance wireless communication systems featuring adaptivity, the provided contribution can be extrapolated to other communication systems with demanding DSP requirements (i.e., systems requiring wide bandwidth and dynamic range; e.g., MIMO channel emulation for power line communications or digital pre-distortion for wireless back-hauling links).

1.1 Thesis overview

The contents of the thesis are organized as follows.

Chapter 2 covers the motivation of the thesis. The different technologies utilized in the modelling and implementation of the PHY-layer are discussed, whose target strictly falls within the realistic prototyping of high-performance wireless communication systems featuring adaptive behaviour. For this reason, a representative diagram of PHY-layer design, implementation and evaluation ecosystem is defined (i.e., the different levels encountered from high-level modelling to fabrication).

Chapter 3 reviews the related work, with the intention to provide a comprehensive taxonomy of relevant literature. The main contributions of the thesis are also detailed and located within the described landscape.

Chapter 4 proposes a design, implementation and verification methodology aimed at providing a performance-efficient implementation of bit-intensive wireless communication systems featuring adaptivity. The methodology covers all the required steps from the definition of the system requirements and high-level modelling to the comprehensive and realistic evaluation of the resulting prototype, using a heterogeneous hardware-setup with close to real life operating conditions. Moreover, it are discussed the multitude of challenges to which the system designer is deemed to face, in many different levels, when following the previously introduced methodology. The latter aims at providing system-level design principles which along with the design work flow constitute a complementary companion that helps to accomplish a successful performance-efficient applied research roadmap.

Chapters 5 and 6 provide two indicative use cases, focusing on the innovation provided in the digital realization of the PHY-layer of advanced wireless communication systems. The aim of those chapters is twofold. First, they serve as a proof-of-concept of the proposed design methodology. More importantly, those RTL design principles constituting a principal contribution of this thesis are fully illustrated therein.

Chapter 7 is closing the thesis by analysing the results and extracting useful conclusions, while pointing future work lines and extensions of the presented work.

Chapter 2

The PHY-layer development ecosystem

Motivating the innovation in digital design

Empowering the constant evolution of the wireless communication systems requires inter-sectorial and cross-sectorial novel solutions. Yet, when strictly focusing on the PHY-layer, innovative solutions are needed at all different levels: theoretical conceptualization, modelling, design and implementation. The identification of these heterogeneous innovations is what this chapter intends to outline. The message that is conveyed is the fact that a long path lays across the proposal of a novel algorithm and its efficient digital realization.

2.1 Introduction to the PHY-layer development ecosystem

Figure 2.1 provides a graphical overview of the different approaches commonly utilized to design, implement and validate novel PHY-layer solutions. Evidently, the development-cycle time highly depends on the selected approach, that could serve different objectives: ranging from a rapid evaluation of known or novel concepts, to an extremely long development process when targeting a final-product. As the complexity of the proposed solutions augments (i.e., high-performance and adaptivity), more optimization is required for its digital implementation. Consequently, a more complex design, implementation and validation method is required. Furthermore, while the conceptual and behavioural modelling of PHY-layer algorithms related to wireless communication systems naturally starts by using High-Level Programming Languages (HLPLs) and a computer-based simulation, it is widely accepted within the experimentally-driven research community that the complete validation of innovating DSP algorithms is only achieved when the latter are implemented and tested using baseband and RF laboratory equipment that generates and processes signals, which are subject to real-life hardware constraints and impairments (i.e., including a physical or hardware-emulated signal propagation channel) [Caban et al., 2011].

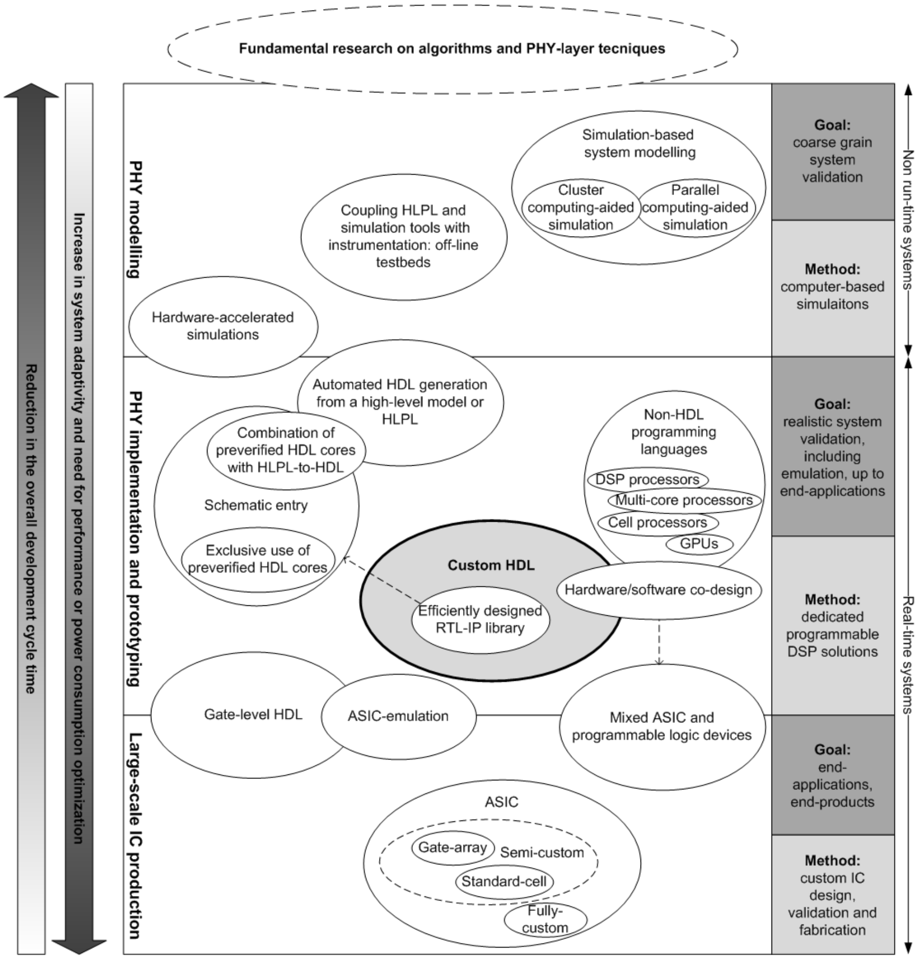

Fundamental research on algorithms and PHY-layer tecniques
Reduction in the overall development cycle time
Increase in system adaptivity and need for performance or power consumption optimization
PHY modelling
PHY implementation and prototyping
Large-scale IC production
Simulation-based system modelling
Cluster computing-aided simulation
Parallel computing-aided simulation
Coupling HLPL and simulation tools with instrumentation: off-line testbeds
Hardware-accelerated simulations
Automated HDL generation from a high-level model or HLPL
Combination of preverified HDL cores with HLPL-to-HDL
Schematic entry
Exclusive use of preverified HDL cores
Non-HDL programming languages
DSP processors
Multi-core processors
Cell processors
GPUs
Custom HDL
Efficiently designed RTL-IP library
Hardware/software co-design
Gate-level HDL
ASIC-emulation
Mixed ASIC and programmable logic devices
ASIC
Gate-array
Semi-custom
Standard-cell
Fully-custom
Goal: coarse grain system validation
Method: computer-based simulaitons
Goal: realistic system validation, including emulation, up to end-applications
Method: dedicated programmable DSP solutions
Goal: end-applications, end-products
Method: custom IC design, validation and fabrication
Non run-time systems
Real-time systems

Figure 2.1: PHY-layer design, implementation and validation ecosystem.

2.1.1 Fundamental research founded on computer-based models and simulations

The innovative ideas impulsing the progression of wireless communications have their origins on fundamental research. In the case of the PHY-layer, the novel signal processing solutions are typically born as computationally demanding mathematical models, which in many cases require of unprecedented processing capacity. Consequently, the real-world conception of such solutions results prohibitive even when state-of-the-art technology is employed. Hence, in many cases a gap exists between fundamental and applied research. At the same time, the ability to verify the capacity gains of novel DSP algorithms and techniques, or the performance of new wireless communication standards, is one of the main research and development drivers of both academic and industrial entities. Therefore, fundamental research studies based on computer simulations, that describe in a high-level algorithms or entire systems, have been traditionally adopted as the principal method to rapidly estimate the achievable gains of the proposed solutions.

The capital features of software-based modelling (algorithmic or system-wide) indeed constitute the most adequate method to evaluate the proposed ideas at an early stage of technology-driven applied research. This is mainly due to the inherent facility provided by HLPLs to describe complex mathematical models, with numerous pre-verified software libraries and tools made available to the system designer. The utilization of the universally prominent C/C++ general purpose languages consists one of the most emblematic HLPL-based DSP modelling approaches. Some indicative open-source initiatives can be found in [AQU, SPU, DSP], whereas [IPP] provides a commercial General Purpose Processor (GPP)-based DSP library. Nonetheless, MATLAB® [MAT] probably constitutes the de-facto HLPL and mathematical modelling environment utilized by the signal processing research community. It is because MATLAB provides the means for rapid verification of signal processing algorithms and systems in a user-controlled cross-platform environment. Additionally, an ever-growing set of add-ons, open-source code and pre-compiled libraries facilitates the high-level design and modelling of complex systems. Within the most commonly utilized MATLAB extensions lies the Signal Processing Toolbox™ [SPT], providing an exhaustive collection of notable industry-standard DSP algorithms. The versatility of MATLAB has also made it one of the standard interfaces between computers and high-end testing, emulation and signal generation instruments.

The utilization of HLPL-based DSP libraries permits the researcher to concentrate on the conception of innovative PHY-layer solutions, as the complicated aspects related to its digital realization can be abstracted away. Another essential benefit of system modelling based on computer simulations is its relatively low development cost, both in temporal and economic terms, providing by this way an ideal environment for the rapid evaluation of complex solutions with inexpensive hardware equipment. Similarly, standard constructs of HLPLs can be utilized to describe the DSP model, not demanding the lengthy acquisition of expertise in the utilization of additional programming languages or tools aimed at constructing low-level hardware architectures (e.g., Hardware Description Languages - HDLs). All the previous advantages are substantially augmented when considering the flexibility provided by any software-based development (e.g., compilation time is short compared to low-level programming languages). The modelled solutions can be modified or extended without a notable change of the development cost. Moreover, complex and governable scenarios can be considered with a similar cost (e.g., large number of users and fully-controlled evaluation conditions, such as the channel model, the Signal-to-Noise Ratio - SNR - or the interference perceived by each mobile receiver). Likewise, the considered scenarios can also be easily modified or augmented.

The above provides a solid basis for computer-based algorithmic exploration or system-wide experimentation, constituting by this way an essential vehicle that enables fundamental research. At the same time the versatility of simulation-based high-level modelling environments is effectively used as an indispensable part of the applied research development cycle, that targets innovating prototypes of systems that are subject to real-life constraints. However, this particular use of HLPL-based mathematical modelling environments reveals a series of limitations which, generally speaking, are related to their finite capacity to deal with computational intensive data processing and other limitations on modelling or accounting for mixed signal effects and impairments. Hence when the numerical complexity of the PHY-layer algorithms increases, it is required to make assumptions and simplifications in the mathematical models, in an attempt to address the mismatch of the required computational capacity and simulation time. While the latter allows the analysis of the theoretical performance gains achievable by the proposed solutions, basic parameters that critically affect the operation and the physical realization of wireless communications are omitted in an idealized simulation environment. This may occasionally result in overoptimistic conclusions (e.g., in terms of performance, required processing architecture or estimated power consumption). A short description of some common assumptions and simplifications, and their impact on the realistic validation of PHY-layer algorithms and systems, follows:

- **Unlimited resources:** high-level software models do not usually account for the amount of baseband signal processing resources that will be required to physically implement the system under test, leading to an ideal validation scenario, where unlimited computational capacity is assumed. This additionally prevents a precise analysis and quantification of the achievable gains, since the DSP-computing architecture remains undefined. Furthermore, the computation latency of the required DSP calculations is also depending on the defined architecture. The lack of consideration of the previous aspects may affect the evaluation of algorithms, even when the digital realization of the system under test is not the principal target. Indicative examples are the analysis of performance or energetic gains, since they are both strongly related to the low-level DSP architecture and implementation technology (e.g., on-chip dedicated memory versus off-chip shared memory).

- **Unlimited precision:** HLPL-based models usually provide highly accurate double-precision numerical environments, leading to floating-point system models that neglect many of the constraints encountered in real-world implementations. In particular, the precision-loss introduced when translating a floating-point system model, built with a HLPL, to fixed-point logic, targeting a specific baseband processor technology, are often underestimated or completely ignored. As a result, the impact of some standard operating conditions or inherent functional features on the performance of an algorithm or an entire system may not be properly evaluated (e.g., noisy channel estimations), which in turn may lead to severe degradation of the global performance of the (wireless) communication system. Likewise, the target technology is limiting the maximum attainable precision (e.g., maximum bit-width provided by the Analog-to-Digital Converter - ADC - or Digital-to-Analog Converter - DAC - stages).

- **Idealized channel conditions:** it is quite common to assume a flat fading channel, with independent and identically distributed fading at each antenna, or simplified (quasi-)static channel models which are considered at simulation-time during the early stage of algorithm exploration. Furthermore, usually it is assumed perfect knowledge of the channel at both the transmitter and receiver sides. The mentioned environmental simplification and channel-knowledge idealization strongly impair the accurate assessment of the selected algorithms suitability and the confident analysis of the computational complexity of a digital realization, which shall fulfil the performance requirements of the system under test. This is especially crucial in those scenarios that consider mobility and, thus, rapid variations of the channel propagation conditions. An illustrative case is the channel estimation at the receiver: channels featuring aggressive fading demand an increased arithmetic computation, often leading to a design-time trade-off between implementation complexity and performance. An accurate definition of this delicate trade-off requires an extensive algorithm selection phase, which is highly dependant on the considered scenario (e.g., channel model, SNR conditions).

- **Ideal signal:** it is not uncommon to encounter in the literature (see Chapter 3) algorithmic studies where the model's input signal or test vector is generated in the same simulation environment and, thus, failing to account for certain natural operating conditions, limitations and features of real-life hardware equipment that critically affect the received signal. In this context, it is often assumed the reception of perfectly synchronized signals, although for instance in OFDM-based systems, Fast Fourier Transform (FFT)-window misalignments have a direct impact on the performance of the receiver. Other potentially significant signal impairments, such as In-phase and Quadrature (I/Q) gain and phase imbalances, inaccuracy between the sampling clocks of the transmitter and receiver in respect to the ideal sampling frequency, Local Oscillator (LO) drifts, random phase noise due to LO instability and Carrier Frequency Offset (CFO), also tend to be neglected. This both affects the estimated performance and simplifies the implementation complexity: on the one hand, one of the key components with a critical impact on the performance of the baseband logic, the Digital Front-End (DFE - i.e., responsible for vital operations of the receiver, such as the symbol detection or the control of the power-gain applied at the ADC stage), is not taken into account. On the other hand, the RF front-end is also neglected, abstracting away the hardware non-idealities of real-world systems (e.g., attenuation suffered by RF components, additional noise originated by the RF equipment, antenna coupling, Direct Current - DC - levels).

- **Control plane:** real-world implementations are usually composed of specialized computing blocks working in parallel, that need to intercommunicate among them, and with other basic system resources (e.g., off-chip memories, buses), in a synchronized manner. Therefore, robust control units, managing the operation of the processing blocks, must be designed to ensure the correct operation of the system. Such control units tend to be complex entities, often requiring a non negligible amount of hardware resources and, likewise, of considerable design, implementation and verification time. This part is naturally not present in software simulations, thus omitting the implementation overhead of the control plane. The relevance of this issue is particularly important when the behaviour of the system needs to be dynamically adjusted, according to the response of external parameters. Hence, not only there will be the need to exchange control data between the transmitting and receiving entities, but it has to be delivered in a timely manner and with minimized overhead. If the latter is not properly dimensioned, analysed and designed, it may heavily influence the expected gains introduced by an innovating PHY-layer algorithm.

Therefore, in many occasions what is missing in the evaluation cycle of innovating PHY-layer algorithms is the emulation of system-wide functional parameters that are subject to realistic physical conditions, inherent to the operation of real-life hardware devices, equipment and instrumentation, as well as realistic channel propagation and SNR conditions. Therefore, in order to assess in exact terms the performance of separate algorithms or the entire PHY-layer of a communication system, it is crucial that the simulation model under validation takes into account as many effects, system specifications and physical impairments as possible. The previous is an essential step to enable both the efficient implementation of novel solutions and a closer-to-reality analysis. Such a detailed model requires a substantially additional computational capacity, which is related to the complexity of the proposed PHY-layer solutions. This is especially true, when considering bit-intensive wireless communication systems, which by default feature elevated computation requirements that are due to the wide baseband bandwidth, the number of antennas, the considered communication-scheme and the time-varying DSP functions.

2.1.2 Realistic PHY-layer modelling

Three main approaches are utilized to improve the previously discussed HLPL-based system modelling, as detailed in the following.

Parallel techniques for simulation-based PHY-layer modelling
The procurement of enriched PHY-layer models focuses on improving the computer-based simulation performance. The required innovation relies on the provision of a massive parallel computation capacity. In recent years, a clear tendency towards parallel computation has been adopted to confront the physical constraints posed by the ever increasing operating frequencies that a single GPP could reach. Numerous solutions have been proposed from clusters of computers to multi-core processors, attaining parallelism at different levels (i.e., bit, instruction, data or task level). The features of the existing HLPLs have been augmented to attain an optimum computing-resource exploitation, enabling the provision of high-performance without loosing portability and scalability. An indicative example is the standardization effort procured by the Message Passing Interface (MPI) [Gropp et al., 1999], which enables parallel computation for the most commonly used HLPLs (i.e., C, C++, Java and so on). Nonetheless, two principal issues need to be addressed to accomplish massive parallel computing. First, advanced parallel processing hardware solutions need to be assembled, engineered or even prototyped. A whole range of solutions exists towards this end, mainly based on standard commercial computers; from a nearly inexpensive single machine including multiple processors (or a multi-core processor) to an extremely costly large cluster of computers, featuring a dedicated high-throughput network (jointly with its associated control-software). A second important issue is the creation of parallel-executable code, which requires the use of precise programming skills that exploit the parallelism encountered in the hardware processing solutions. Failure to provide source code with adequate

parallelism constructs, that respect the compiler and final processing-solution specifications and limitations, results in sub-optimal executables, which may even under-perform compared to non-parallel HLPL and single-processor (single core) solutions. Although in those cases the code is still based on common HLPL-constructs, novelty is required in the Electronic Design Automation (EDA) tools to facilitate the generation of complex PHY-layer models based on a parallel-computing architecture. A notable solution to abstract away the specific parallel-computation coding is provided by the MATLAB Parallel Computing Toolbox$^{\text{TM}}$ [PCT]; this allows the computation and data-intensive models to be executed on multi-core processors, GPUs and clusters of computers in a user-friendly environment. Furthermore, it permits to combine both previously detailed parallel-HLPL approaches; indicatively, optimum custom code (e.g., written in C/C++) can be called from the true high-level MATLAB-generated model, when the latter cannot fully take advantage of the parallel-computing resources.

In spite of the enormous computational capacity attainable with parallel computing solutions, HLPL-based PHY-layer modelling still features a restricted ability to produce realistic conditions for the considered signals. The latter becomes more important when taking into account that, for the successful and correct validation of MIMO systems under time-varying mobility conditions, it is required a solid understanding of the channel propagation characteristics and specifications. Many wireless channel models are typically considered in HLPL-based simulations whose goal is to emulate the propagation conditions and eventually provide estimates of the attainable MIMO channel capacity. Despite their complexity, even the most sophisticated models make many assumptions and ignore common propagation effects (e.g., refraction, diffraction and reflection loss) or correlations among the different antenna elements. Consequently, the utilization of such models can result in important deviations in the MIMO channel capacity estimates [9].

Off-line testbeds

A popular alternative allowing the verification of PHY-layer algorithms under more realistic operating conditions is by assembling hybrid experimental testbeds, that combine real-time processing and off-line software-based post-processing (denoted as off-line testbeds henceforth). This rapid-prototyping approach provides an improved testing environment that partially addresses some of the issues encountered in HLPL-based PHY-layer simulations. In this sense, off-line testbeds offer a trade-off in terms of cost and time for the rapid verification of the functionality and performance of novel PHY-layer algorithms. Such platforms make use of commercial testing, measurement and signal generation equipment (e.g., Vector Signal Generator - VSG - instruments equipped with arbitrary waveform generators), capable of producing real-time RF signals. Then, the VSGs are configured with user-generated vectors representing the output of a baseband transmitter (i.e., produced by a HLPL-based software model) and eventually produce a real-time RF signal that is transmitted, using either antennas or a direct cable connection (e.g., the output of the VSG is connected to the input of a digital oscilloscope). In other more advanced off-line testbed setups where receiver-mobility scenarios are considered, the output of the VSG is connected to the input of a RF channel emulator where static or mobile channels are emulated with real-time hardware. Moreover, other instruments that combine signal generation and channel fading are also used towards this end. Certain additional laboratory equipment might be utilized to control other important parameters (e.g., White Gaussian Noise - WGN - signal generators can help setting the desired SNR conditions, by adding a noise signal to the received Intermediate Frequency - IF - signal). At the receiver side the signal either passes through an RF front-end that downconverts the signal to IF or through an RF IC that makes a zero-IF conversion and provides directly the baseband signal. This signal after the ADC conversion is stored in large buffers (e.g., utilizing an FPGA device) and retrieved in order to be post-processed off-line. If a static channel is considered then, at the receiver, a digitally controlled Programmable Gain Amplifier (PGA) is used to empirically adjust the incoming signal to the maximum observable dynamic range. However, in case a multi-tap channel with (high) mobility is considered, then the absence of a post-ADC Automatic Gain Control (AGC; responsible to automatically adjust the incoming signal to the full dynamic range by configuring in real-time the PGAs) and of a signal detection and synchronization processing block (which are deterministically defined by the format of the incoming signal) is typically forcing to apply a high back-off safety margin (empirically set after observing for a long period of time that the incoming signal does not saturate the ADC). This apparently results in a non negligible loss of dynamic range (SNR), which in turn may affect the performance of the receiver under evaluation. On top of the realistic channel-related conditions, the received signal also includes hardware-introduced constraints (e.g., limited ADC precision, RF non-idealities and losses). The captured close-to-real-world signals are used as inputs to a HLPL-based simulation, which facilitates the realistic validation of the baseband signal processing algorithms of the receiver (e.g., MATLAB high-level model of the system). A graphical overview of a generic off-line prototyping testbed setup is provided in Figure 2.2. Notwithstanding, off-line prototyping is still inheriting various constraints that apply to the HLPL-based simulations presented earlier. Specifically, given its simulation-based nature, the utilized HLPL-modelled PHY-layer is usually not fully accounting for the limited numerical precision of the baseband (i.e., only the quantization introduced in the ADC/DAC circuitry is included). Furthermore, strong restrictions apply when high-performance systems requiring of a dynamic adaptation of the PHY-layer are simulated using a HLPL-based model (further discussed below). On top of it, software simulations are not accounting for the digital design and control-plane overheads.

This type of testbeds are having off-line HLPL-based simulations running at the two ends (transmitter and receiver). Certain researchers tried to reduce the performance specifications of the systems under development, or apply advanced parallel computing principles at the simulations that run in computer systems with multi-core processors, in order to provide a semi-real-time interfacing of the simulations at the two ends with the real-time instrumentation-based testbed. However, when targeting high-performance wireless communication systems, even with the combination of parallel-computation techniques and non-synthetic signals (e.g., real-world data captures), the simulation time is still very high (i.e., it largely depends on the desired precision and statistical accuracy of the simulation results, the selected bandwidth, the transmission mode and the chosen modulation order [10]). Hence, the number of frames that can be simulated is limited. Moreover, as the physical implementation of complex

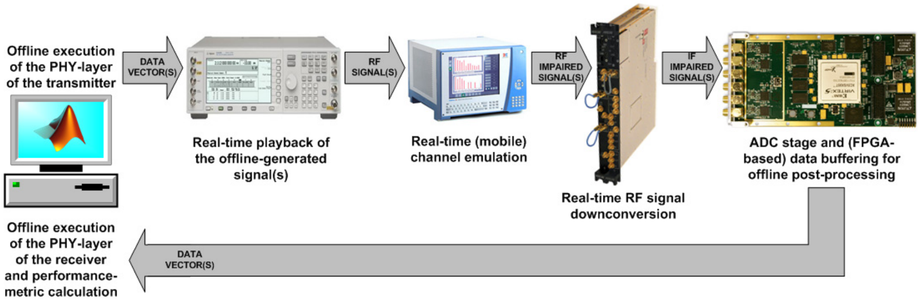


Figure 2.2: Generic off-line prototyping scheme.

baseband blocks that critically affect the global system performance is not considered, the accuracy of the system-analysis may not be substantially improved even when utilizing realistic signals. An indicative example is encountered in the computer-based simulation of the PHY-layer of a wireless communication system (e.g., using MATLAB) which is designed considering that the channel estimation features floating-point arithmetic and favourable SNR conditions of the captured signals; this might result in a significant performance-mismatch between the model and the physically implementable system (i.e., typically realized using fixed-point numerical representations).

Therefore it is practically very difficult, and at the same time quite uncommon, to encounter in the literature PHY-layer developments which accurately model and exhaustively validate dynamic systems when using non real-time processing solutions. This comes as the natural outcome of the increased processing requirements due to the computational load of the advanced DSP techniques being utilized, together with the inherent adaptivity of certain algorithms. A very indicative example that can support the arguments mentioned herein is found in the operation of closed-loop mobile broadband communication systems; the latter require, for instance, a fast adaptation of the transmission scheme according to the current channel conditions, which are communicated through a dedicated feedback link from the receiver to the trasmitter. Essentially, three main approaches have been proposed in the literature to realistically measure the performance of the previously mentioned systems [11] :

(i) Off-line testbeds where the feedback is carried out via a high-speed connection. The round trip times of a typical hardware-setup (e.g., considering an off-line execution of a MATLAB-based PHY-layer for both the transmitter and receiver, a feedback link based on Local Area Network - LAN - and over-the-air transmission) is usually in the order of milliseconds (e.g., 100 ms to 1 second). Consequently, only scenarios where the channel stays constant over long periods of time can be considered (e.g., indoor measurements performed at night, when no human presence is distorting the current channel conditions). Clearly, scenarios featuring constantly moving objects are prohibiting such measurement strategy.

(ii) Off-line testbeds where no feedback is used at all, but a block of data is transmitted for every possible feedback combination and evaluated later on. Of course, this is only realizable in the case of limited feedback. That is only for certain scenarios where the channel remains constant during (at least) the time required to transmit all the possible blocks of data (i.e., considering the time consumed in the feedback propagation, reception and decoding, additionally to that required for the adaptation of the DSP producing the transmitted signal).

(iii) Advanced technology demonstrators where the whole system, including the feedback, operates in real-time. An efficient PHY-layer implementation is required (especially for high-performance MIMO-OFDM systems), as well as a realistic laboratory setup (e.g., FPGA-based DSP-prototyping boards, RF equipment and a channel emulator capable of reproducing mobility conditions for the mobile terminals).

Hardware-accelerated simulations

An interesting alternative to the parallel-computing techniques described earlier is that of hardware-accelerated simulations, commonly denominated as Hardware-In-the-Loop (HIL), which may provide an entry point to the real-time implementation domain. In a HIL simulation-approach, those parts of the system requiring of bit-intensive parallel computation are implemented in hardware (usually a FPGA device), allowing its real-time execution. The remaining DSP functions of the system remain implemented at a software-simulation domain. Shared memories are used to allow the real-time communication between the simulation that runs at a computer system and the FPGA device (i.e., a physical memory is accessible from both the FPGA and the simulation-computer; e.g., the inputs and outputs of the FPGA implementation use embedded Random Access Memory - RAM - residing in the FPGA fabric, which are mapped to a memory space of the computer's RAM through a dedicated firmware interface, included only in specific FPGA boards, and an Application Programming Interface - API - that runs at the computer), providing a high-speed data link, as shown in Figure 2.3. In this case, the attainable throughput of the HW/SW communication is defined by the dedicated communication interface connecting the FPGA and the simulation-computer with the shared memory. Once again, advances are required in the EDA tools supporting the development of HIL-based wireless communication systems. Simulink® [SIM], a MATLAB-based schematic-entry software tool for modelling, simulating, and analysing embedded systems, when combined with System Generator for DSP™ [SYS], a high-level tool for the design of high-performance DSP systems using FPGAs (System Generator essentially interconnects Simulink with the Xilinx FPGA design and implementation EDA toolchain), has probably introduced the most iconic model-based design solution for rapid PHY-layer design (largely discussed in Section 2.1.3). One of the principal features of the mentioned combination of EDA tools is to facilitate a MATLAB/FPGA HIL design and validation strategy. However, although such hardware-accelerated approach could be classified as a partial real-time implementation (or real-time if certain performance specifications are downgraded), the digital realization of the system is not fully contemplated when an EDA tool is used to abstract away the design of the DSP-architecture. HIL can be considered a reasonable approach for applications tailored to implement and validate wireless communication systems, with a relatively low effort and within a limited time-frame. Nonetheless, when accounting for high-performance PHY-layer techniques featuring adaptivity, innovation is

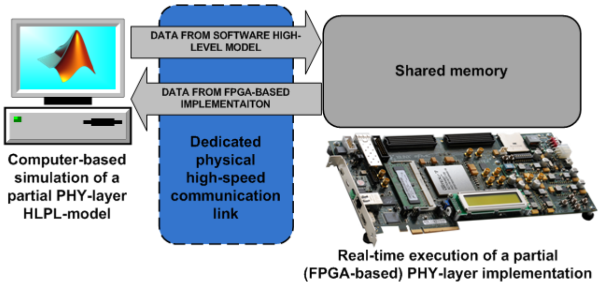

DATA FROM SOFTWARE HIGH-LEVEL MODEL
DATA FROM FPGA-BASED IMPLEMENTAITON
Shared memory
Dedicated physical high-speed communication link
Computer-based simulation of a partial PHY-layer HLPL-model
Real-time execution of a partial (FPGA-based) PHY-layer implementation

Figure 2.3: Generic HIL scheme.

required to provide a further optimized PHY-layer design and implementation approach, permitting the development of robust real-time systems that operate in close to real-life functional conditions.

2.1.3 PHY-layer implementation and prototyping

Prior to examining the different alternatives encountered for real-time system development, it is required to properly define the *real-time* concept.

Real-time signal processing places stringent demands on the design of DSP hardware and software, as it is required to complete predefined tasks within a certain time frame [Kuo and Lee, 2001]. A limitation of DSP systems for real-time applications is that the bandwidth of the system is limited by the sampling rate. Accounting for the Nyquist-Shannon sampling theorem [Shannon, 1949], in order to accurately represent an analog signal digitally (i.e., discrete-time), two conditions must be met. First, the analog signal must be limited in bandwidth, f_M. Second, the sampling frequency, f_s (i.e., ADC-stage), must fulfil the condition $f_s \geq 2 \cdot f_M$. Therefore, in a real-time DSP system the signal processing time, t_p, must be less than the sampling period, $T = \frac{1}{f_s}$, in order to complete the processing task before the new sample comes in. This real-time constraint limits the highest frequency signal that can be processed by a DSP system: $f_M \leq \frac{f_s}{2} < \frac{1}{2t_p}$. Clearly, the longer is t_p, the lower is the supported f_M. Even when accounting for the ever-increasing capacity of state-of-the-art DSP-specialized hardware, there is always a limit to the processing that can be performed in real-time. The latter becomes more apparent when the cost of the system is taken into consideration, given that it is usually required a trade-off between cost and system performance. Hence, many applications simply cannot be utilized because of economical constraints, even if more complex hardware approaches could provide the required computation potential.

When facing the implementation of real-time bit-intensive PHY-layer solutions which require dynamic DSP-adaptation, in response to the current channel-state condition, the principal approaches are classified in two broad categories: Hardware Description Language (HDL)-based and non-HDL designs. HLPLs are used in non-HDL implementations, but consider the efficient utilization of the hardware resources provided by advanced specialized processors, which enable the real-time execution of bit-intensive DSP algorithms. On the contrary, the HDL-based method aims at describing the digital logic of the system at RTL: that is to say, defining the flow of data (i.e., digital signals) between hardware registers (i.e., memory) and the logical operations between them (i.e., processing units) by means of a specialized programming language. The resulting description is finally utilized for FPGA-based or ASIC implementations. The Verilog and Very High Speed Integrated Circuit (VHSIC) HDL (VHDL) are the most widespread HDLs, with the second being utilized for the implementation objectives of this thesis.

The inherent complexity of novel PHY-layer developments poses the application of innovating digital design techniques and requires increased design efficiency to take maximum advantage of the capacity of the target technology, which is ever-growing in an exponentially manner, as predicted early by Moore's Law [Moore, 1965]. The selected implementation technology is finally defining the reachable system performance (e.g., in terms of power consumption and/or data-rate). Each implementation alternative discussed hereafter could result more adequate for different target use cases and development-time (or budgeting) requirements. Nonetheless, both HDL and non-HDL PHY-layer implementations share the following features:

- **Implementation flexibility:** different applications can be implemented with no additional hardware costs (e.g., by reprogramming the FPGA device). Nevertheless, this flexibility comes at the cost of implementation efficiency loss when compared to custom silicon developments, which are programmed at fabrication-time and cannot be modified thereafter (besides minor changes of their firmware).

- **Re-usability:** libraries of hardware-proven designs, known as IP cores, or parts of previously developed designs, can be directly re-used, leading to shorter design cycles and improved implementation quality.

- **Design portability:** the utilization of a programming language to describe the DSP functionality provides an efficient means of migrating between similar technologies (e.g., between FPGA devices or DSP processors). In other words, the resulting code is not necessarily bound to a given device architecture or processor technology. Evidently, though, a predominant technique to augment the DSP design efficiency, as described in the previous point, is to re-utilize code and/or provide a lower level of detail to the design (i.e., instantiate physical hardware resources by using special constructs provided in the utilized specialized programming language). In the previous case, the resulting design is indeed accounting for the low-level architectural details of the implementation technology and, hence, difficults the migration process (i.e., recoding might be needed).

An important aspect to be accounted in the two alternatives is the common utilization of fixed-point arithmetic for the final digital realization. The reduced hardware complexity, when compared to floating-point implementations, provides significant speed and resource-utilization gains which are usually more important than those related to augmented numerical-precision (i.e., high precision is not always required to adequately represent the dynamic range of a system). Nevertheless, the translation of the high-level model of the system (e.g., when using MATLAB, which features floating-point logic), used during the preliminary evaluation stage, to fixed-point HDL code, which will endow its real-time FPGA-based implementation, is a complex task which critically affects the implementation efficiency and the system performance. On the other hand, though, the utilization of floating-point arithmetic facilitates the final implementation (e.g., the tedious translation to fixed-point is avoided, and thus makes straight forward the comparison of the real-time implementation with the high-level model). Therefore, it is important to note that floating-point realizations are not only possible (in the cost of additional processing complexity at baseband), but also necessary for given DSP-applications. Indicatively, specific floating-point arithmetic libraries, IP cores, embedded microprocessors and other dedicated processing components can be used in FPGA devices, to serve the needs of particular applications that require this type of arithmetic operations [12, 13].

To conclude, it is worth mentioning that the MATLAB DSP System Toolbox™ [DST] provides algorithms and tools to help automating the design of advanced PHY-layer techniques for both implementation choices. One of the main features of this environment is the automatic generation of fixed-point C and HDL code from the MATLAB model (including support for the Simulink environment as well).

Non-HDL implementations

The first HLPL-based method aiming at providing an efficient real-time PHY-layer implementation is making use of specialized microprocessors, named DSP processors, which feature an internal architecture and an instruction set optimized for the operational needs of DSP. When compared to GPP-based solutions aiming for real-time PHY-layer developments, the DSP processors feature lower latencies, area and power-consumption, allowing its inclusion in portable DSP-oriented devices. As it is common for HLPL-based approaches, the principal DSP processor manufacturers are providing design tools to facilitate the development of complex systems, denoted as Integrated Development Environment (IDE) [TIC, ANA]. Further EDA-capabilities are enabled when combining those tools with MATLAB. The Embedded Coder® tool [EMC] enables the automatic generation of C/C++ code from a high-level MATLAB model (i.e., MATLAB-code or Simulink model), providing a direct interface with the vendor-supplied IDEs. Notwithstanding, when more efficient DSP implementations are required, custom optimized assembly-code routines (i.e., low-level code) are utilized, due to the inefficiency and limitations of the EDA tools that automatically translate HLPL-based designs or schematic entry models to lower-level code. Indicatively, the main DSP-processor vendors offer DSP libraries of software-functions composed of C-callable assembly-optimized routines [TID, FRE, NXP].

As in the case of advanced HLPL-based PHY-layer simulations, in the real-time systems utilizing dedicated DSP processors there is a clear tendency towards increased parallelism (e.g., inclusion of multiple processing cores) and heterogeneity (e.g., system architectures integrating accelerators such as GPUs or FPGAs). Likewise, from the system designer point of view, innovative EDA tools are required to facilitate the inclusion of parallel computation solutions to the high-level DSP-models aimed at developing real-time systems.

Nowadays there is a handful of powerful processing solutions that can host real-time PHY-layer implementations, which primarily or completely rely on software-based DSP. Similarly to multi-core GPPs, modern multi-core DSP processors may include several independent central processing units. According to

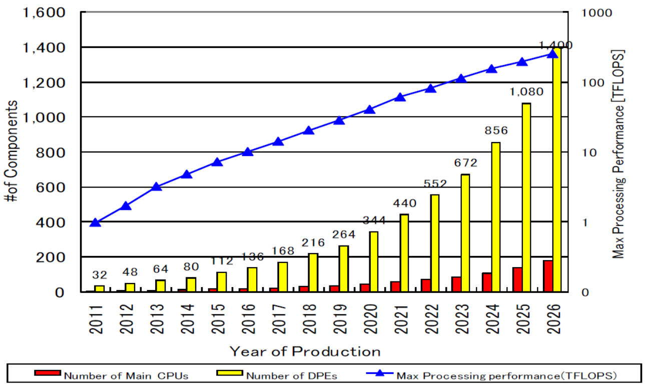


Figure 2.4: Evolution of computational capacity in consumer portable SoCs [ITRS, 2011].

the current evolution of the silicon-based technology, it is expected that hundreds of cores will be integrated in a single chip in the near-future, as shown in Figure 2.4. Additionally, hardware accelerators, usually called co-processors, introduce such processing architectures to efficiently implement specific DSP operations. Probably, the two most indicative technologies favouring a co-processing architecture are the GPUs and the Cell Broadband Engine Architecture (CBEA), usually referred as cell processors.

The highly parallel architecture featured in the GPUs makes them very efficient in applications where processing of large blocks of data is done in parallel. Hence, although originally designed as a co-processor specialized for computer graphics acceleration, GPUs are every time more used, in a versatile way, to perform General-Purpose computation (GPGPU). Their ability to sustain bit-intensive floating-point mathematical operations enables the efficient implementation of advanced real-time PHY-layer techniques using specialized software-code (e.g., C or assembly). Similarly, the CBEA combines in a single chip a GPP and several co-processing elements specially optimized for vectorized floating-point operations, interconnected by means of a dedicated high-performance circular bus. The CBEA architecture provides a significant computing performance for C/C++-coded vector processing applications (i.e., parallel processing of blocks of data). Optimized DSP libraries are available to the system designer from semiconductor vendors [NVI, GEI], as well as from open-source initiatives [ARR, CEL]. Furthermore, the open-source Yellow Dog Linux [YDL] Operating System (OS), is designed to support high-performance computing for both GPUs and cell processors, whilst providing access to the numerous general-purpose open-source code and applications available for the common distributions of the Linux OS.

Hardware/software co-design

The utilization of custom co-processing elements (reprogrammable at run-time) that are coupled with some sort of DSP, GPP or GPU processor and other application specific ICs through high speed on-chip embedded buses, allows the implementation of bit-intensive parts of the system; all these different processing elements reside in the same silicon dye (realizing what is denoted as

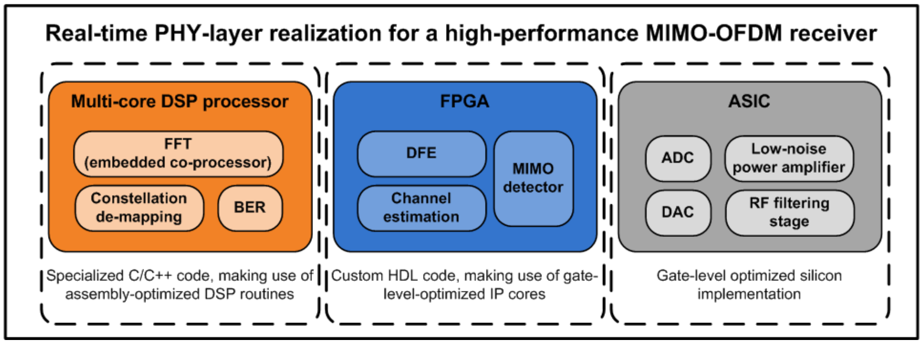
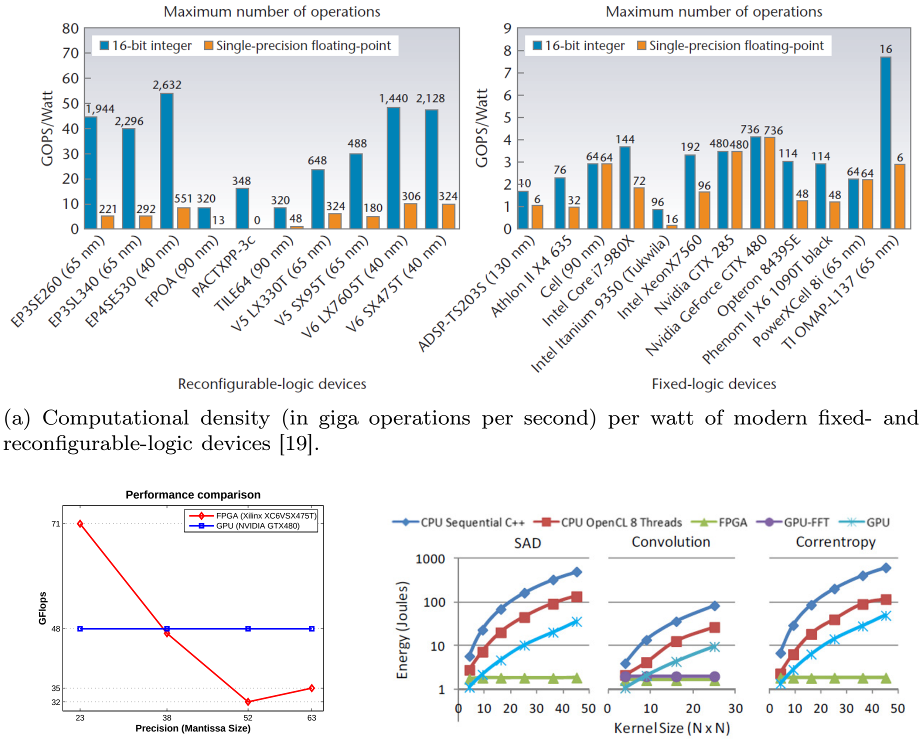

Real-time PHY-layer realization for a high-performance MIMO-OFDM receiver
Multi-core DSP processor
FFT (embedded co-processor)
Constellation de-mapping
BER
Specialized C/C++ code, making use of assembly-optimized DSP routines
FPGA
DFE
Channel estimation
MIMO detector
Custom HDL code, making use of gate-level-optimized IP cores
ASIC
ADC
Low-noise power amplifier
DAC
RF filtering stage
Gate-level optimized silicon implementation

Figure 2.5: Generic example of a real-time PHY-layer implementation based on HW/SW co-design.

System-On-a-Chip - SoC), with the reprogrammable part of them offering a hardware-acceleration prototyping area to those applications that suffer performance bottlenecks when implemented in GPPs or DSP processors. This system implementation approach requires a HW/SW co-design methodology, where part of the system is implemented in software (e.g., HLPL executed on a multi-core DSP processor) and the remaining is implemented by means of custom hardware (i.e., FPGA-based or ASIC implementation), as shown in Figure 2.5. In such approach, innovation is required to enable an efficient HW/SW co-design flow (and EDA tools) able to integrate processing elements of different coding and system design domains (i.e., HLPL or HDL), in order to target different processing component categories (e.g., GPP, DSP, FPGA or ASIC) [14]. Hence, the main challenge of HW/SW co-design is to decide in an optimal way which part of the system is implemented in software and which part in hardware, accounting for the characteristics of the available processing elements (e.g., power consumption, processing capacity, cost and flexibility). This challenge gets more complicated when taking into account that there is rarely a globally optimal implementation technology for a particular application. Therefore, a non-trivial design-space exploration is required to enable an efficient physical implementation of advanced PHY-layer solutions (e.g., preventing designers from choosing inappropriate devices).

Whereas in a hardware-accelerated simulation a GPP is in charge of the off-line execution of a non real-time HLPL-based PHY-layer model, in an implementation based on HW/SW co-design, both the hardware and the software parts are being executed in real-time. As detailed before, the constant evolution of the hardware solutions (i.e., massive parallel computation featuring DSP-optimized architectures) is enabling the real-time execution of demanding software-based DSP implementations. Nonetheless, when the features of these advanced processors cannot fulfil the required design efficiency (e.g., because of their cost, power-consumption or GPP-like architecture), a low-level custom programmable processing solution may provide the answer (e.g., FPGA-based design). An efficiently designed custom hardware solution is featuring parallel stream-computation combined with application-fitted memory structures and functional units. Hence, such application-specific low-level architecture design provides a fully optimized digital realization (e.g., for performance and/or en-



(a) Computational density (in giga operations per second) per watt of modern fixed- and reconfigurable-logic devices [19].

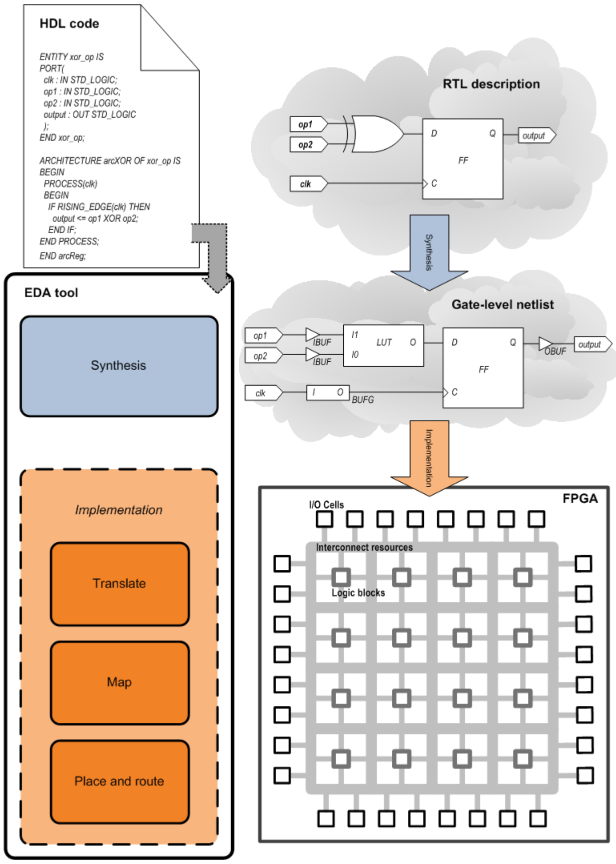
(b) GPU vs. FPGA performance comparison when implementing a dense linear system solver [20].

(c) Energy consumed by GPP, GPU and FPGA devices in a sliding-window application [21].

Figure 2.6: Recent implementation technology comparison studies.

ergy consumption).

FPGAs constitute the ideal entry-point to a HW/SW co-design solution. Modern FPGA devices offer the flexible (and relatively low-cost) means for the realization of efficient dedicated hardware designs requiring of massive parallel computation [15, 16]. Moreover, many recent studies have shown that (when properly utilized) they are capable of outperforming the advanced non-HDL implementation technologies by providing higher performance and/or energy efficiency. A few indicative examples are shown in Figure 2.6. Additionally, the powerful features of state-of-the-art FPGA devices, also allows them to mimic the proposed co-processing architectures. To this end FPGA devices are able to host the HDL version of moderate-performance GPPs and use embedded buses to provide connectivity with on-chip processing cores and on-board peripherals. Furthermore, taking advantage of the physically-available DSP-specialized FPGA-resources (i.e., dedicated Arithmetic Logical Units - ALUs - embedded at silicon-level) the HDL-defined processors can efficiently implement baseband DSP functions [17, 18]. Hence, FPGA devices offer an exceptional solution to efficiently implement advanced PHY-layer techniques, as it will be detailed throughout this thesis.



```
ENTITY xor_op IS
PORT(
  clk : IN STD_LOGIC;
  op1 : IN STD_LOGIC;
  op2 : IN STD_LOGIC;
  output : OUT STD_LOGIC
  );
END xor_op;

ARCHITECTURE arcXOR OF xor_op IS
BEGIN
  PROCESS(clk)
  BEGIN
    IF RISING_EDGE(clk) THEN
      output <= op1 XOR op2;
    END IF;
END PROCESS;
END arcReg;
```

Figure 2.7: Generic HDL design flow and FPGA architecture.

HDL-based implementations

Before entering into a more detailed discussion about the different HDL-based solutions, the basic notions of the HDL-based design flow, and internal FPGA device architecture, will be introduced considering that custom HDL code has been selected as the primary PHY-layer FPGA-prototyping approach in this thesis.

As it can be seen in Figure 2.7 when using a HDL to define a RTL architecture, a top-down design methodology is in fact being adopted. HDLs provide the means to work at a higher level of abstraction than the final low-level description that will be used to program the FPGA device. The first step in this translation process comes with the utilization of a synthesis tool (i.e., EDA software), which produces an optimized logical-level netlist representing the RTL architecture described by the HDL code. During this step, a translation to basic logical entities is performed, containing both logical gates, usually realized by means of Look-Up-Tables (LUTs), and more complex blocks such as memories or DSP-specialized embedded FPGA resources. Then, equivalent circuits are identified to reduce the area and/or increase the performance of the final implementation. The synthesis compilation takes a relatively short time for its execution (although it scales with the complexity of the system), hence, facilitating the exploration of different design decisions (e.g., speed versus area trade-off). Once synthesis is successfully performed, architecture-specific layout tools are required to realize the physical implementation. It is extremely important to note that the synthesis step can produce netlists which are not implementable in the target FPGA, as it includes logical entities instead of its physically-available resources (e.g., performance and implementability are subject to physical constraints explored in the following steps of the FPGA implementation flow).

Different synthesis and optimized implementation algorithms are required for different FPGA architectures (even if they are provided by the same vendor), to fully enable an efficient digital realization of the RTL description. In other words, designers must understand both the target FPGA architecture and the capabilities of the utilized EDA tools in order to leverage an optimum result. Proper configuration and utilization of the FPGA-vendor provided software-tools is a critical factor for efficient FPGA-based implementations, a fact that is not commonly receiving the attention it deserves. Hence, the post-synthesis stages are finally responsible to account for the specific physical resources provided by the target FPGA device, in an attempt to provide an optimized implementation fulfilling the user-defined constraints (e.g., timing constraints fixing the different target operating frequencies for each critical data-path). As a result, a FPGA-programming binary-file is generated, containing a bitstream that once downloaded to the device will finalize the implementation. Generically speaking, an FPGA implementation includes the following stages:

- **Translation and mapping:** this step refines the synthesis-produced logical-level netlist, accounting for those resources physically embedded within the target FPGA device (e.g., number of available LUTs, memories or DSP-slices); as Figure 2.7 shows, they can be roughly classified in I/O, interconnection and logic elements. In other words, the minimal abstraction still remaining after the execution of the synthesis tool is removed at this stage. Consequently, many low-level circuit optimizations are taking place, requiring a considerable longer execution time compared to the synthesis stage.

- **Place and route (PAR):** during this crucial implementation stage, the specific FPGA-resources that will realize the system are selected, accounting for their location within the target FPGA device, as well as their interdependences (i.e., those more conveniently located within the internal network of interconnections are grouped). Then, the physical interconnection between the selected resources is performed accounting for the introduced delays. As these two steps are extremely complicated (especially in dense designs), and also highly dependant on the user constraints, an iterative process is usually executed for this stage. During each iteration, a refined approximation is produced towards the final implementation, which eventually might not be possible (i.e., the user can define a performance-prerequisite upon which, if no successful implementation has been achieved, the process stops providing error-information which aims at helping the designer to resolve the detected issues).

The main HDL-based design and implementation approaches are shortly discussed below.

Automated HDL generation and schematic-entry HDL-based design

As it was the case in the software-based implementations, there has been an extraordinary effort to evolve the HDL-based EDA tools in an attempt to close the gap between design-productivity and the current hardware capacity (i.e., reduced time-to-market and cost). Hence, the principally required innovation for future EDA-based methodologies is to provide an efficient digital realization from a high-level model of the system, where each part is designed using the most appropriate languages and tools (e.g., tackling the previously discussed co-design issues, reusing optimized IP cores, allowing the coexistence of different levels of abstraction and enabling its co-simulation).

In the first steps towards this evolved EDA framework, most of the FPGA board manufacturers are currently offering model-based design flows, trying to approach a broader customer-pool (e.g., software engineers), while at the same time the EDA industry is making all possible efforts to maximize the design and IP reuse. Such model-based design has become extremely popular, frequently allowing the automated generation of HDL code (e.g., VHDL) from a high-level model that is used during the initial system analysis stage (e.g., MATLAB). This High-Level Synthesis (HLS) provides an automatic mapping of the descriptions to reconfigurable processing units, notably shortening the development-cycle time by abstracting away the low-level architectural details. The automatic translation of MATLAB/Simulink models into RTL HDL code (i.e., targeting specific FPGA devices) is widely supported by commercial EDA tools [SMC, SYS, HDC]. Nonetheless, converting a MATLAB model into fully functional HDL code (e.g., for FPGA-based prototyping) requires a considerable effort [22]. When MATLAB code is used for simulations, it allows a high degree of programming freedom; however, this very code has to be (extensively) modified by the designer, in order to enable its automatic translation to a lower-level programming language, such as an HDL one, by using an EDA language translator. This is due to significant language restrictions that apply: only very limited language constructs and specific functions or routines can be automatically translated into an HDL description, and often not with a very efficient result [23, 24]. Additionally, the automated HDL code generation suffers in the translation of the floating-point arithmetic to a limited dynamic range. In other words, it is frequent the requirement to predetermine the fixed-point word length for each intermediate signal required for the RTL architecture. Therefore, the main concern raised is that the MATLAB-to-HDL automatic conversion is not yet mature enough to cover the needs of various implementations; especially those with stringent processing and performance requirements, where specific and hard-to-meet constraints are imposed by the size and the embedded resources of the target device. Hence, these limitations may occasionally render this option unsuitable.

Alternatively, a manual step-by-step conversion allows the indirect translation to HDL. First, the high-level model is exported to C-code (e.g., by using the Simulink Coder$^{\text{TM}}$ [SCD]). Then the C-to-HDL synthesis can be applied using both vendor-provided [SCC, CAC, VSL] or academic-originated [25] software tools. Nevertheless, this automatic hardware mapping process is not always entirely seamless, as the generated C-code might contain unsupported constructs (e.g., pointers, floating-point computation). Furthermore, when targeting FP-

GAs directly from C/C++-based PHY-layer models (i.e., C-based HLS tools), the low-level timing-agnostic HLPL-code used for simulations requires of intensive restructuring in order for the HLS tool to produce satisfactory results (i.e., the initial C/C++ code needs to be rewritten; e.g., representing an architecture specification that can achieve the desired throughput). This code restructuring is a lengthy process, usually demanding of several code-refinement iterations, that requires of a profound knowledge of the target FPGA architecture. The refined C model includes FPGA-specific code optimizations, aimed at improving the timing and providing an efficient use of specific FPGA-resources (e.g., force the utilization of embedded DSP macros, accounting for the bit-widths of the multiply-and-accumulate operations). Hence, the resulting code, along with standard HLPL-constructs, it requires as well the utilization of compiler directives (e.g., pragmas inserted in the C/C++ code) which are necessary to help the HLS tool to produce an efficient RTL description (e.g., to force unrolling of loop-structures or to specify the specific FPGA resources that must be used to implement an array). When such low-level FPGA architectural design details are contemplated from the high-level description, then both the development-cycle time and the design complexity of HLS-based flows are comparable to that of custom HDL code generation [26].

The previous RTL design and implementation approach changes a bit when considering the automated code generation from a Simulink model. Although it is true that it usually requires to restrict the design to the subset of blocks supported by the HLS tool, generally the underlying hardware instances of such blocks have been developed with custom HDL code. For instance, when combining Simulink with the System Generator for DSP, the system designer has indirect access to a library of proprietary IP cores featuring an optimized low-level design (i.e., accounting for the specific resources of the target FPGA device). Additionally, lower level logical entities (i.e., FPGA primitives; e.g., registers, embedded RAM blocks or DSP slices) are also available to the designer. Hence, in this case, the automated-translation can be seen more as a replacement of the high-level description for pre-verified low-level code. Therefore, efficient implementations can yet be achieved by carefully designing the RTL architecture, in a similar fashion to a design based on custom HDL code (but using a schematic-entry EDA tool). Evidently, the latter is true provided that the DSP libraries available to the designer are containing all the components required to implement the desired system. Otherwise, a custom design will be needed: either through a HLPL-based model (latter translated automatically to HDL) or by directly defining its RTL architecture using custom HDL code. Furthermore, some additional design-restrictions are slightly complicating the definition of complex architectures. Indicatively, when using System Generator with Simulink, a single clock signal accompanied by clock-enables is proposed to implement multi-rate systems [XILINX®, 2012c], which is not fitting well for the reutilization of large pre-compiled synchronous designs with hierarchical structure (typical for modular descriptions using pre-verified IP cores).

Custom HDL code

Although it is true that with current EDA tools, even researchers with no prior exposure to digital design techniques can implement DSP algorithms directly from their model-based descriptions (e.g., MATLAB/Simulink), a fully-automated and seamless design and implementation approach still needs to address many issues especially when an optimized design is required in order to yield an efficient PHY-layer implementation (e.g., high-performance MIMO-OFDM systems featuring adaptivity). This is due to the fact that automatically-produced HDL code is usually not as efficient as the custom hand-written HDL one. This difference is becoming a significant factor to be considered when stringent FPGA area utilization conditions apply or when performance and achievable clock frequencies do matter [23]. The modern FPGA devices and the corresponding synthesis tools seem to be rapidly addressing the issues detailed before. This is due to the extraordinary capacities of the new devices in terms of embedded resources (e.g., logic, memories and dedicated DSP-logic) and the significant improvement of the FPGA design and implementation tools. However, given the constantly increasing performance requirements and algorithmic complexity of novel PHY-layer solutions, it is anticipated that the FPGA-based prototyping and the respective EDA tools will be continuously challenged from those factors. Therefore, it is not far from reality to claim that custom-HDL coding will always constitute a stable and reliable way to sort out well-established digital design problems (i.e., dense FPGA designs with compute intensive requirements and hard to achieve timing constraints [27]), even if it is only utilized in a small portion of the design (e.g., complex blocks critically affecting the performance of the system for which a pre-verified IP core is not available). The main point of this argument is that the complexity of such problems is scaled because of the massive amount of FPGA logic, memories and embedded components that need to be addressed. In this sense, custom HDL coding provides the means to control every important aspect of the design, which requires an in-depth knowledge of the low-level RTL architecture. For this very reason, a custom HDL coding strategy that relies on pre-verified IP cores, which feature optimized low-level design, has been mainly utilized in this thesis, to develop the PHY-layer of two different wireless communication systems, as described in Chapters 5 and 6.

Custom HDL code requires a thorough knowledge of the target FPGA architecture and of its associated development framework (i.e., EDA tools). A top-down methodology based on HDL code, features logical or functional abstractions. While this provides an increased productivity (i.e., automatic error-free translation - to a gate-level netlist - of a higher-level model), two major aspects need to be carefully considered, in order for it to produce an efficient physical implementation. First, the expertise of the designer in generating HDL code: because of the evident similarities to an HLPL-based design methodology, system developers and researchers might not understand that, in fact, the HDL code is defining a hardware structure performing the required functionality (and not a set of instructions to be executed in a GPP). Second, the level of detail of the HDL-produced description clearly impacts the quality of the translation process: a very abstract design description might yield poor results, whereas a detailed description drives the decisions of the synthesis tool (i.e., forces the utilization of the instantiated silicon primitives instead of allowing an automatic inference of HDL primitives). As it will be discussed hereafter, in order to produce optimal implementation results for a given target technology, gate-level HDL descriptions are utilized to describe those blocks having a pivotal role in achieving the system performance objectives (e.g., power consumption or data-rate). Finally, IP re-use is also of paramount importance in a custom HDL design to accelerate the development cycle, while minimizing the deign errors.

In fact, the efforts dedicated by both the industry and the academic community to design re-use are ensuring that the provided IP cores are not only featuring an efficient RTL (or even gate-level) design, but guarantee as well a reliable operation. Efficient custom HDL designs naturally provide new DSP libraries (or additional components to already existing ones) to schematic-entry EDA tools.

2.1.4 Large-scale IC production

Efficient Very-Large-Scale Integration (VLSI) implementations of wide-band MIMO-OFDM systems enable high-performance, low-power and low-cost user equipment [Studer et al., 2010]. The main VLSI design and implementation approaches are discussed below.

ASICs

ASICs offer a dedicated hardware solution when mass production is considered, featuring better design security (difficult to reverse engineer their low-level design), better control of I/O characteristics and more compact board design (less complex printed-circuit board, less inventory costs) when compared to FPGAs or other programmable processors [Deschamps et al., 2006]. Likewise, the very high Non Recurring Engineering (NRE) cost (prohibiting its utilization in low-volume production) and the fact that once committed to silicon the design cannot be changed (i.e., the photolithographic masks used as patterns for the fabrication are a main driver of the NRE costs) are the most notable drawbacks. Therefore, when targeting an ASIC it is of paramount importance to achieve an optimum error-free gate-level design.

A large effort has similarly been spent in the automation of ASIC design (i.e., to provide shorter development cycles and lower NRE costs). The latter allows the designers to work at logic gate-level in the so-called semi-custom design. Generally speaking three principal levels of design abstraction are found for ASICs (the first two can be considered within the semi-custom category):

- **Gate-array design:** the silicon waffer utilized to physically fabricate the system contains predefined (arrays of) transistors and other active devices (i.e., logic gates). An ad-hoc computer-aided design tool is used to describe the system, relying on a manufacturer-provided library of components and macros. As a result the tool provides a definition of the required interconnections (between the diffused layers) composing the final implementation (i.e., physical layout). Hence, it is only required to produce the photolithographic masks corresponding to the metal layers (i.e., interconnections, which allow to personalize the master waffer), substantially lowering the final production costs. Nowadays, though, gate-array ASICs have been mostly replaced by FPGAs, which can be seen as their natural technology replacement and evolution (i.e., similar structure allowing a greater design flexibility, capable of attaining a similar performance with a much lower cost). In both cases, a higher utilization of resources implies an increased difficulty in the creation of the interconnection (i.e., it is practically impossible to reach a 100% utilization of their resources).

- **Standard-cell design:** a library of logic components (i.e., gates, multiplexers, adders or even more complex IP cores), encapsulating custom

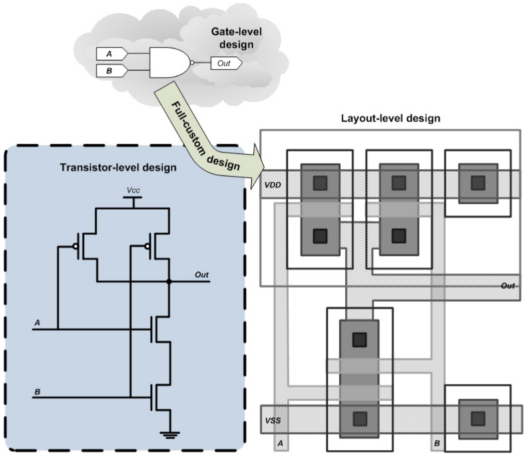


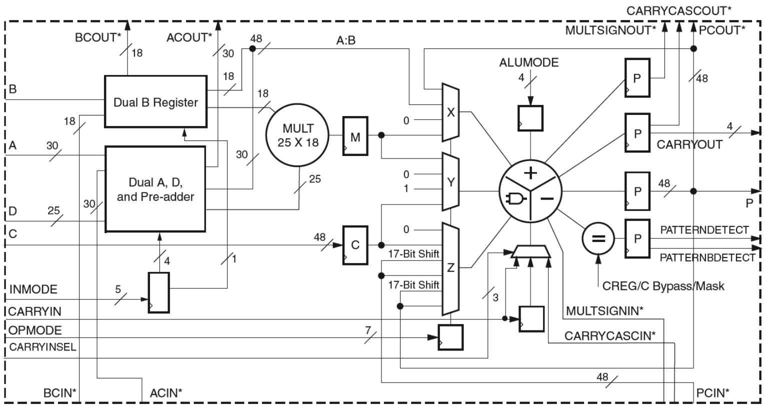
Figure 2.8: Generic full-custom ASIC design flow [Deschamps et al., 2006].

optimized low-level VLSI layout designs, is utilized to conceive the system through a schematic-entry EDA tool or by means of an HDL-based RTL description. Then, specialized commercial IC-synthesis tools are in charge of translating the RTL description to the fabrication layout [DCU, EDI]. An essential benefit of using such abstracted design method is that the standard-cell libraries have been potentially used in thousands of silicon implementations, and thus achieve to substantially lower the design-risk and notably accelerate the development time (i.e., lower NRE costs).

- **Full custom design:** all the layers composing the photolithographic masks utilized for the silicon fabrication are customized (i.e., optimized layout-level design, as shown in Figure 2.8). As it could be expected, a full custom design provides the highest performance, lowest power consumption and the smallest die size, at the cost of increased design time, complexity and price (i.e., slower time-to-market, increased manpower and risk of design-failure). Therefore, this option is only utilized when a highly optimized implementation is required or when no predefined libraries or IP cores do exist for the target design. The most indicative example is the creation of new cell libraries or critical parts of a complex design.

Programmable technologies and ASICs

Given the obvious similarities of the FPGA technology and its associated design flow with the so-called semi-custom ASICs, one of the most common uses of FPGA devices is that of ASIC prototyping, using gate-level HDL descriptions. In other words, following the steps of traditional semi-custom ASIC design-flows, a low-level description is produced accounting for the specific resources provided by the FPGA (i.e., slice level design [XILINX®, 2012b], taking into consideration the available interconnections and the posterior PAR process, which can even be performed manually - at least for the most critical parts of the design). In the case of PHY-layer developments, special attention is brought to the embedded DSP-specialized processing elements (see Figure 2.9), to leverage an optimum utilization of those key processing components. The men-



*These signals are dedicated routing paths internal to the DSP48E1 column. They are not accessible via fabric routing resources.

Figure 2.9: Low-level architecture of an embedded DSP48E1 slice (found in the Virtex-7 FPGA familiy) [XILINX®, 2012a].

tioned gate-level design yields a flexible yet highly efficient realization, that can be thoroughly tested, analysed and optimized before launching its final silicon fabrication. Powerful development platforms featuring several state-of-the-art FPGA devices and, in many cases, other advanced programmable processors (e.g., multi-core DSPs), are interfaced with dedicated high-speed buses and supplied with massive on-board storage capacity and I/O connectivity, to be able to emulate complex ASIC designs.

Finally, the spectacular evolution of the programmable-logic devices (i.e., in terms of capacity and power consumption), jointly with their (relatively) low price, are allowing the FPGA devices to be increasingly used in final end-user applications (e.g., aerospace and defense, medical equipment or automotive communications), jointly with ASICs and other advanced programmable processing solutions (e.g., DSP processors). Hence, it might not be far from reality to claim that the FPGA-technology will play a major role in the digital realization of the PHY-layer of future mobile devices (e.g., by allowing the rapid adoption of new wireless communication standards and its newest releases, without requiring to physically replace or modify the underlying hardware components).

2.1.5 Synopsis

Table 2.1 offers a synopsis of the different design, implementation and validation approaches that have been detailed in this chapter. Indicatively, as more efficiency is required in the design (i.e., to yield a further optimized implementation given a particular target technology and a well-defined application), the complexity of the selected methodology augments as well; likewise, it does its associated development cycle time too.

The combination of Table 2.1 and Figure 2.1 are aiming at providing the reader with the most essential information related to the complete PHY-layer design, implementation and validation ecosystem; that is, for the most common methodologies and technologies utilized for the development of wireless communication systems whose advanced PHY-layer is based on the MIMO-OFDM technology.

Category	**Principal challenges**	**Required EDA innovations**	**Illustrative present solution towards the required direction**
PHY modeling	Enable the efficient utilization of massive-parallel computing solutions	Generation of code making an optimized use of the specifically available computing resources	MATLAB Parallel Computing Toolbox [PCT]
	Achievement of realistic signal conditions	Provision of software-controllable environments resulting in repeatable close-to-real-world signal conditions, accounting for propagation and DSP-introduced delays	Off-line testbeds
	Realistic emulation of dynamic environments		Real-time PHY-layer implementations
PHY implementation and prototyping	Efficient high-level design targeting specialized DSP hardware-solutions	Automatic generation of optimized HLPL-code from a high-level model, accounting for the target processor (i.e., library of optimized assembly routines)	MATLAB Embedded Coder [EMC] (connected to vendor-provided IDEs)
	Enable the efficient utilization of the massive-parallel computing capacity provided by the co-processing architectures	Generation of code making an optimized use of the specifically available computing resources	Vendor-provided [NVI, GEI] and open-source DSP libraries [ARR, CEL]
	Efficient automatic-HDL generation from a high-level model or HLPL	Optimized use of the target FPGA-resources	Commercial HLS tools [SMC, SYS, HDC]
	Efficient and flexible HDL-based design at high-level (e.g., schematic-entry)	Low-level architecture control from abstracted description	Combination of Simulink [SIM] and System Generator [SYS]
		Design freedom	
		Automated integration of parts described at different abstraction levels	
	Efficient design of complex systems targeting hybrid implementation solutions	High-level design allowing the seamless integration of elements of different nature in code (i.e., different abstraction levels and programming languages)	Academic-originated high-level design methodology [14]
		Automatic hardware/software partition from a high-level model	
Large-scale IC production	Efficient high-level design (i.e., minimizing the NRE costs)	Efficient synthesis from HDL code	Semi-custom ASIC design utilizing gate-level HDL code and specialized commercial IC-synthesis tools [DCU, EDI]
		Computer-aided design making use of libraries containing efficient low-level designs	

Table 2.1: Synopsis of the discussed design, implementation and validation approaches.

2.2 Motivation

The main motivation of this thesis resides on the innovation required at the digital design level for being able to efficiently realize the PHY-layer of modern and future wireless communication systems. Taking into account the high performance requirements, computational complexity and high degree of adaptivity of the baseband signal processing of such systems (e.g., MIMO-OFDM closed-loop communication scheme), it is required to include critical novelties, which are not directly related to the proposed DSP algorithmic, but to its actual implementation in a dedicated processing architecture. In order to address the derived challenges, it is fundamental to create high-performance baseband processing engines, based on the FPGA technology, providing architectural solutions to the following issues:

- *Hardware-efficient design:* the real-time implementation of the PHY-layer of advanced wireless communication systems is a challenging task, especially when considering the required computationally-demanding DSP algorithmic (which scales with the bandwidth and number of antennas). To combat the rapid consumption of the finite processing resources encountered in the programmable-logic devices, it is necessary to come up with innovative RTL design techniques leading to an efficient utilization of the underlying hardware means (e.g., embedded DSP-slices and memory blocks). Hence, it is required to intelligently use the proposed algorithms (i.e., to find equivalent mathematical expressions featuring reduced computational requirements) and try to optimize the overall RTL architectures at a very-low level.

- *Resource-sharing:* some common processing blocks are found in the PHY-layer of all wireless communication systems based on the MIMO-OFDM technology (e.g., FFT). Strong similarities can also be found in the DSP algorithmic-calculations of different communication schemes. This needs to be exploited in digital designs featuring an intelligent reuse of computing resources. Indicatively, a resource-shared design enables both a hardware-efficient implementation and helps reducing the overall power consumption of the baseband processing architecture.

- *Optimized parallel architecture:* the performance constraints imposed by the real-time implementation of high-performance PHY-layer solutions, demand for a high degree of parallelism in the baseband processing architecture. This gets more complicated when accounting for elevated processing loads and minimized latency requirements (i.e., high throughput). It is thus essential to define pipelined architectures enabling a fully parallel DSP computation, while leveraging an optimized utilization of the FPGA resources.

- *Adaptive DSP*: a dynamic adaptation of the executed DSP functions (according to the perceived channel conditions) is required in various end use cases (e.g., closed-loop schemes, cognitive or software-defined radio). The real-time implementation of such dynamic processing architectures adds a top-up complexity to the design and implementation of processing-demanding PHY-layer schemes. This is particularly important considering that, in most of the cases, a limited time budget is available to analyse and reprogram on-the-fly the functional processing components that need to be adapted.

- *Flexible control plane:* all the previous points require of a well designed control plane to ensure the timely inter-block and intra-FPGA communications, DSP re-programming and interaction with those entities providing vital information to the operation of the baseband (e.g., dedicated feedback link, MAC-layer or ADC/DAC circuitry).

- *Optimum usage of FPGA design building blocks and primitives:* the efficient FPGA-based realization of bit-intensive PHY-layer solutions requires to optimally utilize the physical processing elements encountered in the underlying silicon-architecture. Hence, the low-level architectural details of the target FPGA technology need to be accounted in the RTL design of the system. For this reason, accurate custom HDL code needs to be generated instantiating the required primitives (both at macro and gate levels), while accounting for their specific characteristics (e.g., numerical-capacity of the DSP-slices, size of the embedded block RAMs or basic logic-components of the standard FPGA slices).

- *Facility to modify the design:* the hectic pace at which the wireless communications advance requires a constant modification, improvement or extension of the implemented PHY-layer techniques. Consequently, the RTL design needs to provide an extensible basis where new functional modules can be added in an incremental way.

A main goal of this thesis is to provide innovating solutions in the digital design of advanced PHY-layer solutions. The efficient implementation and realistic validation of real-time FPGA-based OFDM systems is the target use case. Consequently, a flexible design, implementation and verification methodology is also a desired objective.

Chapter 3

Related work and contribution

3.1 Preamble

The current chapter intends to provide a comprehensive taxonomy of the related literature. The survey of the related work is structured upon the basis of the PHY-layer design, implementation and validation ecosystem depicted in Figure 2.1. The goal of this chapter is to provide a clear picture of the numerous contributions found in the literature regarding the topics considered in this thesis. Before entering into more detail, however, two short issues are requiring further clarification:

- A short definition of basic nomenclature is necessary to provide a fair and ambiguity-free comparison of the differently presented references.
- Specific research groups are used as a primary reference for the quality and depth of their work, and for this reason they are introduced separately.

3.1.1 Definition of basic nomenclature

In the literature, the painstaking process of migrating from a high-level model to a baseband prototype operating in close to real-life conditions is fragmented in various steps. Valuable research contributions can be traced in the work of different authors that target a specific area of the previously presented PHY-layer development ecosystem (see Figure 2.1). Frequently, initiatives with different research goals and extent are ambiguously defined under the same terminology. Furthermore, it is not unusual to find that the efforts conducted on a given isolated development stage are not accounting for the remaining, which results in a partial evaluation of a proposed algorithm. Considering that the principal objective of this thesis is to take into consideration the exhaustive procedure to design, implement and validate high-performance wireless communication systems featuring adaptivity in close-to-real-world conditions, the following literature review is modelled according to this vision.

A terminology definition is given hereafter to ensure a fair comparison of results from different authors and to highlight the contributions of this thesis:

- **PHY-layer or system modelling:** a model comprises code which is not yet adapted to be used beyond a computer-based simulation. Hence, its main use is for rapid evaluation of novel PHY-layer techniques or advanced wireless communication systems. When a real-life implementation of the modelled algorithm or system is the final objective, such abstract representation provides the starting point upon which important design decisions are to be taken. In such cases the model will be refined according to the acquired experimental data to account for real-world hardware and channel propagation constraints. Notwithstanding the model is not bounded by the characteristics of the target technology that is selected for its real-world implementation.

- **Implementation:** this is probably the most ambiguously used term in the literature. By implementation it is referred a development (i.e., code) targeting a specific DSP-processing technology (e.g., FPGA, DSP processor). Hence, this code has been conditioned in accordance to the architecture, resources and specifications of the target processing solution. When FPGA technology is targeted, implementation is considered to be to be the HDL-based development that arrives (at least) up to the synthesis stage and achieves realizable results for the selected FPGA device. Evidently, PAR results are composing a more realistic result, since they are accounting for the specific physical resources provided by the target FPGA device and the user-defined timing constraints (see Section 2.1.3). Furthermore, a major ambiguity is composed by the utilization of the same terminology to refer to developments targeting its real-time operation and those which are not. Specifically, when considering the augmented design complexity associated with the first. Finally, to help differentiating the developments that belong to the VLSI category, it is worth indicating that an efficient FPGA-based implementation aims at providing an optimum utilization of a predefined processing technology (e.g., define a RTL architecture targeting a specific FPGA device). As an indicative example, this thesis targets the efficient FPGA-based real-time implementation of advanced PHY-layer solutions, achieving an optimum trade-off between system performance and implementation complexity.

- **Prototyping:** a prototype refers to a real-time implementation which is integrated onto a hardware platform allowing its realistic evaluation (e.g., a testbed). Hence, the implementation-code is augmented to incorporate board-level integration constructs. In other words, it is required to interface the baseband with additional on-board components (e.g., ADC/DAC circuitry, high-speed buses, I/O, memories and so on). Furthermore, additional laboratory equipment can be utilized to procure the conducted testing with close-to-real-world conditions (e.g., real-time RF signal transmission and reception, realistic channel propagation conditions accounting for mobility). A slight variation of the prototype term is considered for the off-line testbedding approach, since in that case the baseband per se is only modelled. Nonetheless, it is indeed interfaced with a real-world platform. This is typically achieved by using an FPGA device as a large buffer to capture on-the-fly the received signal, which is then parsed and processed off-line in a computer by the corresponding high-level model.

- **VLSI design:** the term VLSI is utilized as an adjective (e.g., VLSI implementation) to differentiate those designs and implementations targeting a custom silicon fabrication (e.g., ASIC). In such case, the design requires to be optimized at low-level (e.g., gate-level HDL), in order to achieve an extremely efficient silicon realization of the custom defined DSP architecture and processing engines. Hence, it can represent a refined step of a previously verified implementation (e.g., FPGA-based).

3.1.2 Research groups used as a primary reference

The work of certain research groups is related in many different ways with the development and applied research goals of this thesis. Moreover, they have a long standing contribution to the fields of interest of this thesis, which for some of them, spans across the entire ecosystem of applied research focusing on the PHY-layer. Their presentation follows alphabetic ordering.

Communications Engineering Department, Universidad de Cantabria (Santander, Cantabria, Spain)
The Santander-based team, under the name of Advanced Signal Processing Group (GTAS in spanish), holds a broad experience in the development of innovative DSP algorithms and its experimental validation, based on MIMO technology [28]. For this reason the group has deployed a FPGA-based prototyping platform, denoted as the GTAS-testbed in this review. The latter is simultaneously enabling both an off-line approach and the real-time implementation of advanced DSP-based systems considering realistically conditioned RF signals [29].

Department of Electrical and Computer Engineering, Rice University (Houston, Texas, United States of America)
Rice University is the birthplace of the popular Wireless Open-Access Research Platform (WARP) [30], an advanced FPGA-based experimental testbed which allows the prototyping of advanced wireless communication systems. The WARP framework provides an open-access repository comprising a basic MIMO-OFDM IP library: the provided baseband designs are oriented towards a model-based design flow (i.e., schematic-entry design), while a software-implementation of the MAC layer is also contemplated [31].

Department of Electronics and Systems, Universidade da Coruña (A Coruña, Spain)
The contributions found in the literature produced by the Group of Electronic Technology and Communications (GTEC in spanish) of Universidade da Coruña are quite diverse. The common denominator of their results is the experimental-nature of the presented work. The group has also developed a flexible FPGA-based testbed [32]. During the review, we will identify this platform under the name of GTEC-testbed. A key feature of the testbed is its distributed multilayer software architecture, designed to facilitate the access and control of the platform [García-Naya et al., 2010].

Department of Information Technology and Electrical Engineering, Eidgenössische Technische Hochschule (ETH) Zürich (Zurich, Switzerland)

The ETH has provided the academic community with fundamental knowledge about the efficient hardware implementation of advanced wireless communication systems and innovative PHY-layer techniques. It is one of the pioneers that took early initiatives that marked and paved the way in the field, such as its relevant contribution to the Multiple-Access Space-Time Coding Testbed (MASCOT) project, which set the basis for the design and digital realization of Multi-User (MU) MIMO wireless systems [Burg et al., 2006], a topic that still remains challenging nowadays. Although a MU MIMO testbed relying on the FPGA technology was developed and utilized for the prototyping of advanced systems [33], the proposed designs were optimized at gate-level in a true VLSI implementation approach [Studer et al., 2010].

Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut (HHI) (Berlin, Germany)

The HHI is a research institute devoted to the entire spectrum of telecommunication technologies and forms part of the large German Fraunhofer research foundation. A fact that makes HHI different is its long standing collaboration with leading industrial entities [34]. As a consequence their experimental investigations are build upon large-scale real-world wireless communication infrastructure deployments. A flagship initiative is the Berlin LTE-Advanced testbed [35], a true MU MIMO-OFDM HW/SW co-design development with advanced PHY-layer features, which also includes the MAC layer [36].

Institute of Communications and Radio-Frequency Engineering, Technischen Universität (TU) Wien (Vienna, Austria)

This Vienna-based group has established experience in the experimental evaluation of advanced PHY-layer communication schemes utilizing realistically-conditioned RF signal transmission. Although the versatility of the group has produced contributions at all levels of the previously discussed ecosystem and its surroundings [37], its most relevant contribution is probably found in the field of off-line testbedding. The Vienna MIMO testbed [38] provides a flexible rapid prototyping approach to examine MIMO algorithms and channel models in close-to-real-world conditions [39]. It should be also underlined the strong link that exists between TU Wien and the Forschungszentrum Telekommunikation Wien (FTW), a leading institution for the research and development of future wireless communication systems. Indicatively, both partners had a fundamental role in the MASCOT project (which was coordinated by the FTW).

Mobile Communications Department, EURECOM (Sophia Antipolis, Alpes-Maritimes, France)

EURECOM is an Information and Communication Technologies (ICT) research center founded by Institut Eurécom, organized in a consortium composed by both relevant universities and industrial partners. Its versatile and extensive capacities can be envisaged in the OpenAirInterface initiative, a flexible open-source HW/SW development platform in the area of digital radio communications. The OpenAirInterface is aimed to be utilized both for real-time DSP implementations [40] and as a large scale wireless emulation platform [41].

3.2 Review of related work

As already mentioned the classification of the literature review is is structured around the ecosystem presented by Figure 2.1. It is important to note, that due to the focus of this thesis, the encountered research based on computer simulations is only considered in those cases that account for realistic PHY-layer models and signal conditions.

Moreover, the relation of the work presented in this thesis with the presented literature is shortly discussed within the review. The complete details of the original contribution of this thesis are provided in Section 3.4.

3.2.1 Advanced PHY-layer modelling initiatives

Computer-based simulations

The flexibility and low development cost of the high-level models used in the software simulations allow them to be used in a wide variety of cases where novel PHY-layer solutions are required. For instance, the power minimization achieved by jointly configuring relevant PHY-layer and hardware parameters is realistically analysed in [42]. Hence, the provision of realistic simulation models has been a long-term objective of many authors within the academic community.

Moreover, another vertical relationship is encountered in the reutilization of results and measurements produced by experimental off-line testbeds in computer-based PHY-layer simulations. For example, the knowledge and experimental data obtained with the Vienna MIMO testbed has facilitated the development of the Vienna LTE simulators [10], which constitute a MATLAB-based link and system level simulation framework. More specifically, the link-level simulator allows the rapid software-based evaluation of advanced PHY-layer techniques, focusing on the MIMO-OFDM LTE technology. The simulator presents adequate characteristics to enable the utilization of its code in an off-line prototyping approach. This is achieved by using separate transmitter, channel and receiver sub-functions. Taking into account the computational capacity requirements, the high-level model of the baseband makes use of the MATLAB Parallel/Distributed Computing Toolbox, combined with highly optimized C functions that assist vectorization operations. A principal objective of this open source simulation framework is to offer a reference tool in the research community, that will allow researchers to fairly compare their algorithms. Some indicative use cases of this simulation framework are interference management in a MU MIMO scenario [43], optimal power distribution among pilot and data subcarriers [44] and Signal-to-Interference-Plus-Noise Ratio (SINR) prediction in a multi-cell scenario [45].

Similarly, an extension to the widely used NS-2 network simulator in order to include a cooperative OFDM PHY-layer is proposed and verified based on a real-time implementation over the WARP testbed in [46]. Furthermore, in [47] the large scale wireless emulation platform provided by the OpenAirInterface is used to simulate a MU LTE-based frequency reuse scenario.

In another case, a channel model derived from experimental data is used by the GTEC group to realistically simulate the influence of residual transmit impairments on the performance of non-linear MIMO receivers [48].

Relation to the work presented in this thesis: the encountered literature which falls within the computer-based simulations category is summarized in Table 3.1. This type of research provides high-level models of diverse PHY-layer solutions, using experimentally obtained data. The utilization of MATLAB-based PHY-layer models accounting for hardware constraints and realistic signal conditions constitutes an initial step of the design, implementation and verification methodology presented in this thesis (Chapter 4).

Reference	**Use case**	**Considered specifications and operating conditions**
[42]	Power minimization by optimally configuring both the hardware and PHY-layer	Flat fading channel
[43]	Interference-alignment techniques for MU MIMO-OFDM systems	Perfect channel knowledge 1.4 MHz bandwidth (BW)
[44]	Optimal power distribution between data and pilot subcarriers	1.4 MHz BW
[45]	SNIR prediction for OFDM systems in multi-cell scenarios	Flat channel response 14 kHz BW
[46]	Distributed cooperative protocol for the interference management in MU OFDM-based systems	IEEE 802.11 (i.e., 20 MHz BW, using 64 subcarriers)
[47]	Evaluation of a MU LTE-based frequency reuse scenario	$\geq$10 ms channel coherence time 5 MHz BW
[48]	Evaluation of the effects of RF impairments in the performance of a 4x4 MIMO receiver	PHY-layer not tight to a standard Custom frames composed of 1000 QPSK symbols

Table 3.1: Synopsis of the literature relying on computer-based simulations.

Off-line prototyping

The massive parallel processing requirements of simulating advanced wireless communication systems, especially when realistic channel conditions are considered, allow the computer-based models only to synthetically emulate certain operating and functional conditions. This often results in a system/algorithm validation that accounts only for a subset of those parameters that critically affect the performance of the simulated PHY-layer. A possible solution to this problem is to obtain real-world data by setting up experimental testbeds, which are based on high-end instrumentation equipment. The provision of realistic conditions for the considered signals is the principal objective of the off-line testbeds, which can be seen as a different level of system validation whose goal is the rigorous assessment of novel PHY-layer solutions. Off-line testbeds maintain the flexible and agile spirit of software simulations, and also benefits from coupling the simulated baseband part of transmitters and receivers with instrumentation and realistic channel propagation conditions. An indication of the flexibility featured in off-line testbeds is their ability to realistically assess the performance of a wide variety of PHY-layer schemes: Time Division Duplex (TDD) zero-forcing linear precoding for OFDM-based systems [49], standard-compliant MIMO-OFDMA mobile WiMAX receivers [50, 51] or Coordinated Multi-Point (CoMP) for LTE-based systems [52].

A popular initiative towards this end is provided by the openly accessible WarpLab framework [53]. WarpLab provides the necessary software that facilitates the direct interaction with the WARP hardware from the MATLAB workspace. This typically includes the generation of frames which are then transmitted using the WARP hardware or capture of signals using the FPGAs populating the WARP boards. Indicative prototyping results are the full-duplex scheme [54] and MU Beamforming (MUBF) [55] for WLAN systems.

Similarly, off-line prototyping efforts are also reported by both HHI [56][1], related to interference management in a frequency reuse scheme, and ETH [57], regarding the compensation of residual transmit signal impairments.

A major reference in the off-line-based rapid prototyping methodology is found in the Vienna MIMO testbed. The latter was used to evaluate a CFO compensation scheme for Single Input Single Output (SISO) WiMAX systems [58] and the performance assessment of MIMO High-Speed DownLink (DL) Packet Access (HSDPA) [59]. An extension of this testbed [60] was the fruit of the collaboration between the Austrian and the GTEC groups. Another relevant example of this collaboration is the experimental validation of an Adaptive Modulation and Coding (AMC) scheme for MIMO WiMAX systems [61].

The GTEC-testbed also provides an interesting off-line prototyping platform. For instance, in [62] the performance of a Bit Interleaved Coded Modulation (BICM) scheme is evaluated under realistic channel conditions. Similarly, the GTAS-testbed is utilized in [63] for the experimental characterization of 4x4 MIMO channels. Furthermore, a very powerful off-line prototyping platform is resulting from the connection of both testbeds. Two illustrative use cases are the evaluation of advanced channel estimation schemes for MIMO systems [64] and the validation of interference-alignment techniques for MU MIMO systems [65].

Finally, a measurement-based infrastructure recently deployed by TU Wien aims at assessing the real-world performance of the IEEE 802.11p standard [66].

Relation to the work presented in this thesis: the encountered off-line prototyping research, summarized in Table 3.2, is providing signals with close-to-real-world conditions (i.e., accounting for both hardware-originated impairments and real-world channel conditions). Once captured, the signals can be utilized to provide a realistic functional framework to the computer-based simulations of innovating PHY-layer algorithms and systems. However, off-line testbeds can only be used to study certain operating conditions which typically excludes high mobility channels and realistic closed-loop systems. An off-line prototyping approach was also utilized in the preliminary development stages of this thesis. Furthermore, during the testing, debugging and verification phases, realistic signal captures were also used to assess the precision or the performance of the implemented DSP functions.

Reference	**Use case**	**Considered specifications and operating conditions**
[49]	Precoding for TDD systems on reciprocal-channel conditions	SISO
		Indoors over-the-air transmissions
		8-point FFT
		Standardless PHY-layer
[50]	Compliant mobile WiMAX transceiver PHY-layer	SISO
		7 MHz BW
		Cabled RF-connection

(Continued on next page)

[1] The authors describe it as a HIL platform, but given that it features a MATLAB-based PHY-layer, we considered it an off-line testbed (see Section 2.1.2 for our vision of HIL).

Reference	**Use case**	**Considered specifications and operating conditions**
[51]	Methodology to analyse MIMO signals using a single RF receiver	2x1 MISO
		10 MHz BW
		Cabled RF-connection
[52]	LTE-based MU CoMP scenario	SISO
		10 MHz BW
		Outdoors over-the-air transmissions
[54]	Full-duplex scheme over WLAN	SISO
		20 MHz BW, 64 subcarriers
		Indoors over-the-air transmissions
[55]	MUBF over WLAN	4x1 MISO
		20 MHz BW, 64 subcarriers
		Indoors over-the-air transmissions
		Channel emulator (mobility)
[56]	Interference-management techniques in a frequency reuse scenario for MU OFDM-based systems	2x1 MISO
		256-point FFT
		Standardless PHY-layer
		Perfect synchronization
		Cabled RF-connection
[57]	Compensation of residual transmit impairments for WLAN systems	4x4 MIMO
		20 MHz BW, 64 subcarriers
		Cabled RF-connection
[58]	Compensation of residual CFO for WiMAX systems	SISO
		256-point FFT
		Outdoors over-the-air transmissions
[59, 60]	Performance assessment of MIMO HSDPA	From SISO to 4x4 MIMO
		Limited feedback
		Outdoors over-the-air transmissions
[61]	AMC scheme for MIMO-OFDM WiMAX systems	1x2 SIMO and 2x2 MIMO
		5 MHz BW
		Limited feedback
		Outdoors over-the-air transmissions
[62]	BICM scheme for MIMO-OFDM systems	2x1/3x1 MISO and 2x2 MIMO
		2.9867 MHz BW
		Standardless PHY-layer
		Indoors over-the-air transmissions
[63]	Experimental characterization of MIMO channels	4x4 MIMO
		20 MHz BW, 128-point FFT
		Standardless PHY-layer
		Emulated mobility
		Indoors over-the-air transmissions
[64]	Advanced channel estimation techniques for MIMO systems	2x1 MISO and 2x2 MIMO
		7 MHz BW
		Standardless PHY-layer
		Indoors over-the-air transmissions
[65]	Interference-alignment techniques for MU MIMO systems	2x2 MIMO
		1.4 MHz BW
		Standardless PHY-layer
		Indoors over-the-air transmissions
[66]	Vehicular communications for IEEE 802.11p OFDM-based systems	SISO
		20 MHz BW, 64 subcarriers
		Outdoors over-the-air transmissions

Table 3.2: Synopsis of the off-line prototyping references.

Hardware-accelerated simulations

The valuable experimental data acquired with off-line testbeds come at the cost of certain limitations; these are associated with the inherent limited processing capacity of HLPL-based PHY-layer models, which also feature a limited ability to implement PHY-layer systems which require high adaptivity due to rapidly changing channel conditions. An indicative example of such limitations is encountered in closed-loop communication systems. Indeed, in order to ensure the coherence of the channel conditions experienced by the captured signals, with the off-line executed feedback mechanism (usually noted as limited feedback in the literature), the experimental conditions require to be fully controllable and reproducible during the complete validation cycle. Indicatively, in [11] it is detailed how the data-vectors originated by all possible feedback combinations are generated off-line, transmitted and captured in real-time. Then, the generated data-captures are evaluated off-line by a MATLAB model. Hence, it is required for the channel to remain constant during the transmission of all data-vectors (i.e., large coherence time). In the case of over-the-air transmissions, quasi-static channel conditions need to be considered (e.g., testing in an indoor environment at night, when no human presence is expected or, in the case of outdoor testing, only non-variant controlled environments can be considered). Similarly, when a channel emulator is employed, a complex laboratory-setup is additionally needed to precisely control and reproduce a set of predefined mobile channel conditions (e.g., predetermine the precise time-instant at which the transmissions are executed). Nevertheless, even under the described circumstances, the non-deterministic behaviour of the analog hardware and the lack of control over the noise and the real-world wireless channel realization, makes it impossible to fully reproduce or trigger specific test conditions [67].

As it can be deduced from the previous, a carefully designed real-time PHY-layer implementation is required to realistically evaluate adaptive DSP techniques. Notwithstanding, given the elevated cost to provide efficient real-time implementations of the complete PHY-layer of modern wireless communication systems, hybrid solutions based on the previously discussed off-line prototyping methodology have been proposed. A representative example is found in [68] where a real-time implementation of the synchronization stage of a 2x2 MIMO-OFDM receiver is combined with an off-line baseband HLPL-model. A complementary effort is found in [69] where a MATLAB channel model is used to test an FPGA-based OFDM transceiver. Furthermore, a HIL deployment is used by the GTEC group where a real-time FPGA-based channel emulator is utilized to realistically assess the performance of a MATLAB-modelled mobile WiMAX PHY-layer [70].

Relation to the work presented in this thesis: the hardware-accelerated simulations, summarized in Table 3.3, represent preliminary efforts towards the real-time implementation of advanced PHY-layer solutions, which forms part of the main objective of this thesis. Furthermore, the encountered references are making use of two indicative BWA technologies (i.e., LTE and WiMAX) which consist the underlying technologies of the use cases detailed in this thesis (Chapters 5 and 6).

3.2.2 PHY-layer implementation and prototyping efforts

Non-HDL programming languages

The utilization of HLPLs to implement the PHY-layer provides higher design flexibility compared to a full custom HDL design flow. For this reason, a

Reference	Use case	Considered specifications and operating conditions
[68]	Preliminary LTE-based receiver prototype featuring a partial real-time implementation of the DFE	2x2 MIMO
		10 MHz BW
		Indoors over-the-air transmissions
		Partial real-time implementation (non-detailed RTL-design)
[69]	HIL-deployment featuring an FPGA-based OFDM transceiver and a simulated channel model	SISO
		64 subcarriers
		DFE not included
		Non-detailed RTL-design
[70]	HIL-deployment featuring a real-time FPGA-based channel emulator and a simulated mobile WiMAX PHY-layer	SISO
		512-point FFT
		Channel emulator (mobility)
		Off-line baseband

Table 3.3: Synopsis of the encountered hardware-accelerated PHY-layer simulations.

software-based PHY-layer serves better the requirements of Software Defined Radio (SDR)-based systems. Indicatively, in [71] a GPP is used to implement a spectrum-sensing controller of a frequency agile Wireless Sensor Network (WSN).

An ideal vehicle to deploy GPP-based SDR systems is provided by the OpenAirInterface wireless emulation platform. Its analog front-end is interfaced through a FPGA-based board with PC-based hosts making use of a real-time extension of the Linux OS. By this way, software-based (limited) versions of different PHY-layers - resembling those of mobile WiMAX and LTE - can be executed in real-time jointly with the MAC and above layers. The inherent flexibility of the previous setup allows the rapid realization of a plethora of SDR-oriented scenarios. Two indicative examples are: a single-frequency mesh network based on a distributed MIMO receiver [72] and an opportunistic cognitive transmission scheme for OFDM-based broadband systems [73].

The general-purpose design of the utilized processors is the principal performance-limiting factor of software-based PHY-layer implementations. Nonetheless, the inclusion of parallel computing architectures and techniques partially helps overcoming this bottleneck, at the cost of a less flexible programming style and, thus, requiring a longer development time to implement the necessary software parallel processing routines (e.g., it features low-level optimizations for a given processing architecture).

A popular parallel computing solution are the cell processors. In [74] a WiMAX BS transceiver is implemented in a server including two cell processors. Similarly, a cell-based platform developed at HHI is utilized to conduct a preliminary capacity-analysis towards a 12x12 MIMO-OFDM LTE+ receiver [75] and to implement a MIMO turbo receiver for LTE-Advanced [76].

The GPUs provide another processing solution that features intensive parallel computation capacity. This has been the target technology that was selected to implement advanced MIMO detectors by both the Vienna and the Rice University groups. More specifically, a preliminary implementation of a sphere decoder is described in [77] and a GPU-based reconfigurable soft-output detector is detailed in [78].

Relation to the work presented in this thesis: the encountered non-

HDL initiatives, summarized in Table 3.4, include real-time implementations of high-performance PHY-layer solutions relying on high parallelism, which constitutes a main driving factor of both the methodology and the use cases described in this thesis. Furthermore, the commented references are considering the main BWA technologies, as well as frequency-reuse scenarios, as their principal application which coincides with the use cases presented in Chapters 5 and 6.

Automated HDL generation from a high-level model or HLPL

HLS EDA tools allow to exploit the inherent flexibility provided by FPGA devices while keeping the low development cost of HLPL-based modelling. Hence, a FPGA-based implementation relying on automatically generated HDL provides an excellent vehicle for rapid prototyping. This is due to the fact that the tedious RTL design is abstracted away at the cost of losing implementation efficiency. Indicatively, the authors in [79], use both HLS tools and a custom RTL design flow to implement a WiMAX transmitter.

The previously mentioned loss of RTL implementation efficiency due to the non-optimal functionality of the HLS tools, does not constitute a serious issue for preliminary development analyses. A relevant example is given in [80] where a spectrum sensing technique for OFDM-based SDR transceivers is evaluated. Similarly, a hybrid Simulink-based design, combining both automatic Matlab-to-HDL and pre-verified System Generator blocks, is used to evaluate a reduced-complexity MIMO Maximum-Likelihood Detector (MLD) in [81].

It is worth mentioning that those implementation efforts solely based on the utilization of a library of pre-verified HDL IP cores (e.g., thus featuring an underlying optimized custom RTL design), when following a model-based design approach, have been included in the *schematic entry HDL-based design* category. This is because it has been considered that, although a HLS tool is indeed employed, the attainable implementation efficiency is bound to the high efficiency of the underlying optimized HDL processing blocks comprising the library (which in fact can be comparable to that of a custom HDL design approach).

Relation to the work presented in this thesis: the presented work related to automated HDL generation, summarized in Table 3.5, constitutes the first indications of FPGA-based digital system design approach, that is closer to the to the design and development framework utilized in this thesis.

Hardware/software co-design

When both the MAC and PHY layers are meant to be jointly designed and implemented or when the predefined architecture of the previously detailed processors does not result convenient to implement a given bit-intensive DSP technique (e.g., limited capacity) a HW/SW co-design could provide provide the required solution. Those parts of the system requiring a greater flexibility are implemented as software that is executed in a general purpose processor (i.e., non-HDL), while a custom RTL design acts as a hardware-accelerator for those components for which an efficient implementation is indispensable (e.g., high-performance or low-power). Three main HW/SW co-design categories have been identified in the literature.

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided benchmarking metrics	Complete system implementation?	Functional/behavioural verification
[71]	Spectrum-sensing controller of a frequency agile WSN	COTS 802.15.4 equipment	GPP-based implementation	No benchmarking is provided	No (isolated PHY-layer processing block)	Computer-based simulation Over-the-air transmissions
[72]	Single-frequency mesh network based on a distributed MIMO receiver	2x2 MIMO 5 MHz BW, only sync. and pilot symbols LTE-based PHY-layer	Real-time GPP-based implementation	No benchmarking is provided	Yes	Computer-based simulation Over-the-air transmissions
[73]	Opportunistic cognitive transmission scheme for OFDM-based broadband systems	2x2 MIMO 5 MHz BW WiMAX-based PHY-layer Limited feedback	Real-time GPP-based implementation	No benchmarking is provided	Yes	Computer-based simulation Over-the-air transmissions
[74]	MIMO-OFDM transceiver	2x2 MIMO 10 MHz BW WiMAX standard	Real-time cell-based implementation	Workload analysis	Yes	Computer-based simulation
[75]	MIMO-OFDM receiver	12x12 MIMO 20 MHz BW LTE standard	Cell-based implementation	Latency Throughput	No (isolated PHY-layer processing blocks)	No functional validation is detailed
[76]	Turbo MIMO-OFDM receiver	4x4 MIMO 20 MHz BW Iterative detection-decoding LTE-Advanced standard	Real-time cell-based implementation (with FPGA-based up and downconversion)	Throughput	Yes (although synchronization and AGC are omitted)	Computer-based simulation
[77]	Fixed-complexity MIMO sphere decoder	4x4 MIMO 2048 subcarriers 64-QAM	GPU-based implementation (non-optimized code)	Speedup vs a multi-core GPP	No (isolated PHY-layer processing block)	No functional validation is detailed

(Continued on next page)

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided benchmarking metrics	Complete system implementation?	Functional/behavioural verification
[78]	Reconfigurable soft-output detector	2x2 and 4x4 MIMO 5 MHz BW 4/256-QAM	GPU-based implementation	Throughput	No (isolated PHY-layer processing block)	Computer-based simulation

Table 3.4: Synopsis of the reviewed non-HDL implementation references.

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided implementation results	Complete system implementation?	Functional/behavioural verification
[79]	Comparison of HLS-based and custom RTL implementation for an OFDM transmitter	SISO 20 MHz BW, 256-point FFT WiMAX standard	Custom HDL combined with automated HDL generation from MATLAB (i.e., intermediate C-code)	Synthesis results with RTL design details	Yes	HDL simulation (ModelSim)
[80]	Spectrum sensing for OFDM-based SDR transceivers	SISO 20 MHz BW, 64 subcarriers WLAN standard	Automated HDL generation from a Simulink model, combined with an open-source SDR PHY-layer	No implementation results and no low-level design details	Yes	Laboratory testing
[81]	Reduced-complexity MIMO MLD	4x4 MIMO 16-QAM	Hybrid Simulink-based design (automated HDL and pre-verified IP blocks)	Synthesis results with no low-level design details	No (isolated PHY-layer processing block)	HDL simulation (ISE)

Table 3.5: Synopsis of the encountered PHY-layer implementations utilizing automated HDL code generation.

The first type of commonly encountered HW/SW co-design research works is featuring a FPGA-based PHY-layer implementation which is controlled by a software-based MAC layer. An illustrative example is given in [82] where a programmable OFDM-based PHY-layer targeting SDR applications is presented.

The WARP platform provides, among other important contributions in the field, a notable HW/SW co-design approach. For instance, in [83] the modulation of a WLAN-based PHY-layer is controlled by the MAC layer depending on the observed packet-loss. More interestingly, fundamental HW/SW co-design notions enabling an efficient PHY/MAC cross-layer design are reported in [84], enabling the design and implementation of a distributed cooperative MAC and OFDM-based PHY layer.

A similar architecture is presented in [85] by the GTEC-group, where it is described the development of a 4x4 MIMO-enhanced transceiver based on the IEEE 802.11a standard, featuring a custom FPGA-implemented baseband processor tightly integrated with a hybrid MAC-layer implementation (i.e., specialized GPP interfaced with a HDL-based hardware accelerator).

Another slightly different HW/SW co-design approach that is usually encountered in the literature provides an hybrid implementation of the PHY-layer by utilizing both HLPL-based and FPGA-realized processing blocks (e.g., combined with a software-based MAC layer). An illustrative example of such approach is given in [86] where different HW/SW partitions are analysed to improve the decoding rate of a MIMO sphere decoder.

Similarly, a reconfigurable mobile WiMAX PHY-layer implementation is presented by the GTEC-group, where an FPGA is in charge of the most intensive DSP blocks and a DSP processor implements the subchannel-related operations [87].

In most of the cases, the advanced wireless communication systems implemented by the HHI-group are based on a well defined HW/SW co-design strategy, which takes advantage of the virtues of both FPGA and DSP processor technologies [88]. Although a custom design of the baseband is mentioned in many of the encountered papers, the details of the implemented PHY-layer are never uncovered. It seems reasonable to relate the black-box nature of the publications to the strong industrial-support that researchers have received (e.g., IP-right issues). Nevertheless, this by no means is diminishing the reported experimentation based on complex real-world deployments, since the resulting measurement-based metrics are providing fundamental information for the construction of future wireless communication technology. The Berlin LTE-Advanced testbed, comprising a high-performance MU MIMO-OFDM LTE PHY-layer, combined with the MAC and internet protocol layers, offers an ideal platform to develop proof-of-concept of advanced real-world experimental setups. Some indicative developments that were built upon this testbed include the full frequency reuse (and extended coverage) by employing indoor relaying [89], interference management through CoMP joint transmission [90] and interference-aware scheduling in the DL [91].

Finally, the last identified HW/SW co-design approach is centred around the design and implementation of an embedded FPGA-based processor, which could be configured to serve the needs of different DSP applications. A software definition layer is used to efficiently and flexibly realize the required PHY-layer algorithm or system, by providing an architecture-optimized programming code. This type of HW/SW co-design methodology is the one encountered in the OpenAirInterface Express MIMO platform. The latter is using a custom FPGA-based DSP-optimized processor, which is controlled from an embedded GPP (also serving as an interface of the software-implemented MAC layer). This aims at enabling real-time re-programmability to support all the functional requirements of any OFDM-based air-interface; at the same time it achieves the reuse of the baseband processing resources for multiple standards [92]. The capabilities of this custom baseband processor are investigated in [93] by analysing its capacity to implement a 802.11p receiver.

Similar work is presented in [94], where a custom FPGA-based assembly-programmable processing architecture for baseband DSP functionality is implemented and utilized to prototype a quasi-optimal sphere decoder.

Additional references have been identified regarding the design of custom DSP-optimized processors, which are included in the section entitled *efficient design targeting IC-technology.*

Relation to the work presented in this thesis: real-time adaptive and/or configurable PHY-layer implementations are considered in the presented HW/SW co-design references, summarized in Table 3.6. Precisely, the real-time implementation of high-performance systems featuring an adaptive PHY-layer is one of the main objectives of this thesis. Yet, it has to be pointed that the presented use cases feature only a limited (emulated) MAC layer functionality. Furthermore, systems where open-loop and closed-loop transmission schemes coexist have been detailed, which are particularly relevant with the first use case presented in this thesis (See Chapter 5 for more details).

Schematic entry HDL-based design

The combination of Simulink and System Generator has evolved almost as the de-facto standard model-based design environment for PHY-layer implementations targeting FPGA devices.

A very indicative example of this schematic entry HDL-based design environment is offered by the WARP framework [Amiri et al., 2012]. The latter is based on a custom RTL architecture which is defined by utilizing a vendor-provided library of FPGA primitives and more advanced macro-blocks (e.g., FFT), both of which are instantiated in a graphical high-level model together with other software constructs; the HLS capabilities of the EDA tool automatically translate this model to HDL code. Two indicative use cases of the WARP platform implement a cooperative OFDM-based PHY-layer [95] and a MIMO LTE receiver [96]. It should be noted that few design details are usually provided in the encountered references, where the focus is generally laid on the experimental verification of advanced PHY-layer schemes. Nevertheless, the WARP software and firmware repository embraces the open and free software initiative making it possible to verify low-level design details directly from the source code files.

The flexible design features of the MATLAB-based framework that makes use of the Simulink schematic entry environment, which is connected with the Xilinx EDA tool-chain through the System Generator, are able to accommodate a large number of similar system developments. A representative example is found in [97], where the authors implement a MIMO WLAN PHY-layer design favouring resource-reuse, and another in [98], where the GTAS-testbed is used to develop and validate a PHY-layer prototype that combines open-loop and closed-loop MIMO schemes.

Ref.	**Use case**	**Considered specifications and operating conditions**	**Utilized design and implementation method**	**Provided implementation results**	**Complete system implementation?**	**Functional/behavioural verification**
[82]	Programmable MIMO-OFDM PHY-layer for SDR applications	SISO 256 subcarriers Standardless PHY-layer Incomplete DFE Limited emulated MAC-functionality	Simulink & System Generator (custom RTL design)	PAR results with RTL design details	Yes	Laboratory testing
[83]	WLAN-based PHY-layer with a MAC-layer-determined modulation	SISO 20 MHz BW, 64 subcarriers WLAN standard MAC-layer	WARP repository (Simulink & System Generator)	PAR results with no low-level design details	Yes	Laboratory testing
[84]	Distributed cooperative MAC and OFDM-based PHY layer	SISO 10 MHz BW, 64 subcarriers WLAN-based PHY-layer MAC-layer	WARP repository (Simulink & System Generator)	PAR results with no low-level design details	Yes	Channel emulator (mobility)
[85]	MIMO-enhanced 802.11a transceiver	4x4 MIMO 20 MHz BW, 64 subcarriers WLAN-based PHY-layer MAC-layer	Simulink & System Generator (custom RTL design)	PAR results with no low-level design details	Yes	Cabled RF-connection
[86]	Efficient HW/SW implementation of MIMO sphere decoders	4x4 MIMO 16-QAM	Custom HDL	Synthesis results with HW/SW partition details using an embedded processor	No (isolated PHY-layer processing block)	HDL simulation (ISE)

(Continued on next page)

Ref.	**Use case**	**Considered specifications and operating conditions**	**Utilized design and implementation method**	**Provided implementation results**	**Complete system implementation?**	**Functional/behavioural verification**
[87]	Configurable OFDM PHY-layer	SISO	Undefined (but the authors have reported custom RTL designs using Simulink & System Generator)	A multi-FPGA implementation is indicated with no low-level design details	Yes	Cabled IF-connection
		Up to 10 MHz BW				Over-the-air transmissions
		Mobile WiMAX standard				
		Limited emulated-MAC functionality				
[89–91]	Advanced PHY-layer schemes for a MU MIMO-OFDM system: full frequency reuse through indoor relaying, interference management through CoMP joint transmission and interference-aware scheduling in the DL	2x2 MIMO	Custom HDL	No implementation results and no low-level design details	Yes	Computer-based simulations
		20 MHz BW				
		Closed-loop broadband MIMO relaying or Multi-cell CSI feedback or TAS and MRC				Over-the-air transmissions
		Controlled interference				
		Adaptive modulation				Channel emulator (static channel)
		UL implementation				
		LTE standard				
		Limited emulated MAC-functionality				
[92]	Custom DSP-optimized processor for SDR systems	Implementation-capacity analysis targeting a software-based SISO 802.11p receiver (10 MHz BW, 64 subcarriers) [93]	Custom HDL code	Synthesis results with RTL design details	No (custom DSP-optimized processing architecture)	Benchmarking of the custom DSP processor
[94]	FPGA-based programmable processing architecture for baseband DSP	Efficient software-based implementation of a sphere decoder for a 4x4 MIMO WLAN-based system	Custom HDL code	PAR results with gate-level design details	No (custom DSP-optimized processing architecture)	HDL simulation (ISE)

Table 3.6: Synopsis of indicative HW/SW co-design initiatives found in the literature.

Other related examples include a fixed-complexity high-throughput MIMO detector [99], a configurable sphere detector for MU-MIMO SDR systems [100], a soft-output detection with turbo decoding [101], a robust channel estimator [102], MIMO-OFDM based channel coding and interleaving techniques [103] and a dynamic spectrum allocation algorithm for OFDMA systems [104].

Relation to the work presented in this thesis: the encountered schematic entry HDL-based designs, summarized in Table 3.7, are defining custom RTL architectures, often relying on pre-verified IP cores, that flexibly implement different PHY-layer solutions. A principal actor in the methodology proposed in this thesis is the custom RTL design enabling the efficient implementation of high-performance adaptive PHY-layer solutions. However, the developed use cases presented in Chapters 5 and 6, are making use of custom HDL code and not of an automated tool that translates HLPLs to HDL or high-level models (that may include pre-verified HDL processing blocks and pure software functions or routines) to synthesizable HDL code. Indeed, the produced RTL design applied in the use cases presented in this thesis can be considered an OFDM-based IP library that potentially can be reused in a model-based design approach. As it was identified in the work of the authors presented in this section, the application scenarios and schemes developed in this thesis include an interference-management technique and a MIMO-OFDM PHY-layer featuring both open and closed-loop communications.

Custom HDL design

An alternative approach to model-based design is the utilization of custom HDL code which enables a more fine grain representation of the digital circuit and the ability to fully control the fundamental aspects of those algorithms and systems under design. The latter is especially important when dense FPGA implementations are struggling to meet timing constraints.

A very representative example of this design approach is provided by researchers of the ETH in [105], where it is detailed a complete 4x4 MIMO WLAN PHY-layer prototyping effort featuring a hardware-efficient design, mapped to a multi-FPGA implementation and validated by experimental means.

Other similar WLAN-based developments have also been found in the literature. The authors in [106] implement a Space-Time Block Code (STBC)-based MIMO scheme, whereas in [107] a Transmit Antenna Selection (TAS) configuration is developed. A DSP-block re-utilization for a minimized hardware cost is presented in [108] and a spectrum-aware adaptive subcarrier allocation in [109].

Several authors have as a goal to efficiently implement the most advanced DSP techniques through optimized custom HDL code. Some indicative examples are iterative MIMO decoding [110], STBC MIMO decoding [111], autocorrelation-based OFDM signal detection for CR systems [112], reconfigurable two-dimensional pipeline FFT processor for MIMO OFDM systems [113], fixed sphere decoder [114] and 2-phase channel estimation for mobile WiMAX receivers [115].

An important point that has to be underlined is that the extremely demanding bit-intensive computation of the DSP algorithms proposed to implement the PHY-layer of future wireless communication systems, require to employ advanced FPGA design techniques in order to provide efficient digital realizations accounting for the limitations of the latest silicon technology (e.g., provide reduced area, latencies and/or power consumption). Hence, low-level optimized FPGA prototypes provide a flexible, powerful and cost-effective means to realistically analyse the cost of implementing such techniques before entering into a true VLSI implementation. Indicative related work in this sense is a fully-parallel Low-Density Parity Check (LDPC) decoder presented in [116], a low-power synchronization technique for WiMAX systems described in [117] and a high-throughput iterative MIMO detection detailed in [118].

Relation to the work presented in this thesis: the presented references, summarized in in Table 3.8, are using custom HDL code to prototype various advanced PHY-layer techniques. In the same way, a custom HDL code is chosen as the main vehicle utilized in this thesis to detail the RTL architecture of the considered systems. As a consequence, the methodology proposed in this thesis is also following the custom HDL design flow. Furthermore, the presented use cases are corroborating its effectiveness in the design and implementation of high-performance adaptive PHY-layer solutions, similar to the ones considered in the reviewed literature.

3.2.3 Efficient design targeting IC-technology

The last category of related research includes VLSI implementations, which go a step further than the previously presented FPGA prototyping efforts by considering the physical fabrication of the designed circuitry.

The authors in [119] implement in Complementary Metal-Oxide-Semiconductor (CMOS) technology a WLAN transceiver which can be considered an evolution of the one presented in [105]. Moreover, the broad experience of ETH in the field of VLSI design is widely documented in the literature. Some indicative examples include the design of a MIMO sphere decoder [120], a Compressed-Sensing (CS)-based channel estimation for LTE receivers [121] and a Successive Interference Cancellation (SIC)-based MIMO detector [122].

A couple of additional notable VLSI implementation of WLAN transceivers are reported in [123,124]. Similarly, in [125] the ASIC implementation of a high-throughput channel estimator for LTE receivers is presented by the research team in Vienna. Furthermore, many authors have worked towards efficient implementations of advanced DSP architectures. Some representative examples include a power-efficient MIMO detector [126], a MIMO near-MLD with channel preprocessing [127], a self-configuring FFT/Inverse FFT (IFFT) hardware macrocell for OFDM-based systems [128] and a Singular Value Decomposition (SVD)-based MIMO channel estimator [129].

Interestingly, not all the VLSI designs encountered in the literature are making use of custom HDL code. For instance, in [130] a Simulink design using System Generator is combined with ASIC libraries containing bit-true equivalents of the utilized schematic-blocks, to efficiently implement a MIMO sphere decoder. Additionally, as introduced in Section 2.1.3, the combination of specialized C-based HLS tools and low-level optimized C/C++-code (i.e., accounting for the specific architectural details of the target silicon technology) leads to implementation results comparable to those obtained when using custom HDL code. Indicative references that are having as a principal target the VLSI design of advanced PHY-layer solutions are a high-performance MIMO-OFDM LTE receiver [131] and a parallel search-based sphere detectors for MIMO-OFDM systems [132] (from Rice University).

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided implementation results	Complete system implementation?	Functional/behavioural verification
[95]	Cooperative OFDM-based PHY-layer	SISO 12.5 MHz BW, 64 subcarriers WLAN standard	Simulink & System Generator (custom RTL design)	PAR results with no low-level design details	Yes	Over-the-air transmissions
[96]	MIMO LTE receiver	2x2 MIMO 20 MHz BW LTE standard DFE not included	Simulink & System Generator (custom RTL design)	Synthesis results with no low-level design details	Yes	MATLAB simulation (utilizing real measurements from WARP)
[97]	Hardware-efficient MIMO-OFDM PHY-layer	From SISO to 3x3 MIMO 20 MHz BW, 64 subcarriers WLAN standard	Simulink & System Generator (custom RTL design)	Synthesis results with RTL design details Resource saving based on time-multiplexing	Yes	No functional validation is detailed
[98]	MIMO-OFDM PHY-layer combining open and closed-loop schemes	2x2 MIMO 3.5 MHz BW WiMAX standard Perfect CSI assumed (emulated closed-loop)	Simulink & System Generator (custom RTL design)	PAR results with no low-level design details	Yes	Over-the-air transmissions
[99]	Fixed-complexity high-throughput MIMO detector	4x4 MIMO 16-QAM	Simulink & System Generator (custom RTL design)	Synthesis results with no low-level design details	No (isolated PHY-layer processing block)	HDL simulation (ISE)

(Continued on next page)

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided implementation results	Complete system implementation?	Functional/behavioural verification
[100]	Configurable sphere detector for MU-MIMO SDR systems	Up to 8x8 MIMO 4/16/64-QAM	Simulink & System Generator (custom RTL design)	Synthesis results with RTL design details	No (isolated PHY-layer processing block)	HDL simulation (ISE)
[101]	Soft-output detection and turbo decoding for MIMO-OFDM systems	4x4 MIMO 64-QAM	Simulink & System Generator (custom RTL design)	Synthesis results with no low-level design details	No (isolated PHY-layer processing blocks)	HDL simulation (ISE)
[102]	Robust channel estimator for MIMO-OFDM systems	4x4 MIMO 6.25 MHz BW, 1024-point FFT	Simulink & System Generator (custom RTL design)	No implementation results with no low-level design details	Yes	Channel emulator (mobility)
[103]	Channel coding and interleaving techniques for MIMO-OFDM systems	2x2 MIMO 20 MHz BW, 256-point FFT	Simulink & System Generator (custom RTL design)	Synthesis results with RTL design details	No (isolated PHY-layer processing blocks)	MATLAB simulation
[104]	Dynamic spectrum allocation for OFDMA systems	SISO 4 MHz BW WiMAX standard	Simulink & System Generator (custom RTL design)	Synthesis results with no low-level design details	No (isolated PHY-layer processing block)	MATLAB simulation

Table 3.7: Synopsis of the encountered references related to schematic-entry HDL-based designs.

Ref.	**Use case**	**Considered specifications and operating conditions**	**Utilized design and implementation method**	**Provided implementation results**	**Complete system implementation?**	**Functional/behavioural verification**
[105]	Hardware-efficient MIMO WLAN PHY-layer	4x4 MIMO 20 MHz BW, 64 subcarriers Configurable modulation Channel coding WLAN standard	Custom HDL	PAR results with RTL design details Complexity vs performance analysis	Yes	Channel emulator (static)
[106]	STBC-based MIMO WLAN PHY-layer	2x2 MIMO 20 MHz BW, 64 subcarriers Configurable modulation Channel coding WLAN standard	Custom HDL	ASIC-synthesis results with RTL design details	Yes	Over-the-air transmissions
[107]	MIMO WLAN PHY-layer featuring TAS	3x2 MIMO 3.32 MHz BW Configurable modulation Channel coding WLAN-based PHY-layer DFE not included Emulated UL (wired feedback)	Custom HDL	A multi-FPGA implementation is indicated with no low-level design details	Yes	Cabled IF-connection to a multipath fading simulator
[108]	Hardware-efficient MIMO WLAN receiver	SISO 20 MHz BW, 64 subcarriers WLAN standard AGC not included Constellation de-mapping not included	Custom HDL	PAR results with RTL design details DSP-block reuse for a low hardware cost	Yes	HDL simulation (ISE) Laboratory testing

(Continued on next page)

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided implementation results	Complete system implementation?	Functional/behavioural verification
[109]	OFDM-based PHY-layer featuring spectrum-aware adaptive subcarrier allocation	SISO 20 MHz BW, 64 subcarriers WLAN-based PHY-layer AGC not included	Custom HDL	Synthesis results with no low-level design details	Yes	MATLAB simulation
[110]	Iterative MIMO decoding	4x4 MIMO Up to 16-QAM	Custom HDL	Synthesis results (for different iteration lengths) with no low-level design details	No (isolated PHY-layer processing blocks)	HDL simulation (Quartus) FPGA-based system emulation
[111]	STBC MIMO decoding	2x2 MIMO Up to 64-QAM 1024-point FFT	Custom HDL	Synthesis results with RTL design details	No (isolated PHY-layer processing block)	HDL simulation (ModelSim)
[112]	Autocorrelation-based OFDM signal detection for CR systems	SISO 64-point FFT	Custom HDL	Synthesis results with RTL design details	No (isolated PHY-layer processing block)	HDL simulation (ModelSim)
[113]	Reconfigurable two-dimensional pipeline FFT processor for MIMO OFDM systems	Up to 4x4 MIMO Up to 1024-point FFT	Custom HDL	Synthesis results with RTL design details Hardware-efficient variable-length pipelined architecture	No (isolated PHY-layer processing block)	No functional validation is detailed

(Continued on next page)

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided implementation results	Complete system implementation?	Functional/behavioural verification
[114]	Fixed sphere decoder for MIMO OFDM receivers	4x4 MIMO 16-QAM Support for spatial diversity/multiplexing schemes Mobile WiMAX standard	Custom HDL	Synthesis results with RTL design details	No (isolated PHY-layer processing block)	HDL simulation (Quartus)
[115]	2-phase channel estimation for MIMO OFDM receivers	SISO 10 MHz BW Mobile WiMAX standard	Custom HDL	Synthesis results with RTL design details	No (isolated PHY-layer processing block)	HDL simulation (Quartus)
[116]	Fully parallel LDPC decoder for OFDM-based systems	SISO BPSK Irregular codes (mobile WiMAX standard)	Custom HDL	Synthesis results with gate-level design details	No (isolated PHY-layer processing block)	HDL simulation (ISE)
[117]	Low-power synchronization for OFDM-based systems	SISO 64 subcarriers Mobile WiMAX standard	Custom HDL	Synthesis results with gate-level design details Optimized multiplierless architecture	No (isolated PHY-layer processing block)	HDL simulation (ModelSim)
[118]	High-throughput iterative MIMO detection	4x4 MIMO QPSK	Custom HDL	Synthesis results with gate-level design details	No (isolated PHY-layer processing block)	HDL simulation (ISE)

Table 3.8: Synopsis of relevant prototyping efforts relying on the creation of custom HDL code.

Furthermore, an optimized C-based co-design flow, developed at the Interuniversitair Micro-Elektronica Centrum (IMEC, Belgium), combines a custom ASIC architecture with SDR for the efficient implementation of advanced MIMO-OFDM systems [133]. Similarly, the development of custom baseband processors is proposed to provide an efficient yet flexible VLSI implementation of DSP functions and systems. Some indicative examples include a DSP-RAM processor targeting Layered Space-Time (LST) Coded receivers [134], a programmable co-processor for MIMO decoders [135] and a wideband spectrum-sensing processor for cognitive applications [136].

Relation to the work presented in this thesis: the encountered literature related to VLSI design and implementation, summarized in Table 3.9, is focusing on fully optimized digital realizations of complex DSP solutions considering real-world fabrication constraints. This fact requires to efficiently define the processing architecture at a low level (e.g., gate-level HDL design). As already mentioned before, FPGAs are often utilized as a cost-effective programmable-technology for VLSI design evaluation purposes. In this sense the encountered literature shares a common ground with this thesis; this is basically the provision of an optimized FPGA-based implementation of advanced PHY-layer solutions. Indeed the key objective of this thesis is to find the optimum trade-off between design complexity and system performance by considering the limitations of the target FPGA technology. The RTL architecture developed for the case studies of this thesis is not fully optimized at gate level, featuring however limited gate-level optimizations in selected parts of the design to yield the required performance.

3.3 Objectives, assumptions and conditions for the developed case studies

The objective of this section is to highlight certain objectives, factors and conditions that were considered throughout the development and validation stage of this thesis. The goal is to underline how this thesis differs from the related work of other researchers by emphasizing the following points:

- **Efficient RTL design of complete PHY-layer architectures:** the principal goal is the performance-efficient RTL design and real-time implementation of complete high-performance PHY-layer architectures using custom HDL code. In other words, it is considered the design of the minimum set of baseband DSP processing blocks encountered in the PHY-layer of modern wireless communication systems. An important parameter of this design is that it explicitly considers the specifications and performance limitations of the utilized hardware-setup. Nevertheless, several operating and functional conditions apply while others are emulated as detailed hereafter:

  – In the presented use cases (see Chapters 5 and 6) only the DL communication was designed and implemented.

  – An emulated UpLink (UL) communication is utilized for the closed-loop schemes to enable the connection between the receiver and the transmitter. On-board and inter-board buses are hosting this communication link which features a non-variable latency for the complete feedback-cycle (i.e., transmission, propagation, reception, decoding and adaptation of the DSP at the transmitter). Nevertheless, the overall system design jointly takes into account both the feedback communication timing and the processing latency for the realistic implementation of the adaptive communication scheme. This also implies that it is not assumed perfect CSI at the transmitter for the DL communication.

  - Only limited and emulated MAC-layer features were considered in order to reduce the already increased design, implementation and verification complexity. Hence it has only been included the provision of a bit-sequence composing the user-data to be transmitted and the subcarrier allocation (PHY-layer adaptation) according to the received feedback-information.

- **DSP algorithm optimization accounting for the required FPGA-resources:** the design goal that was persistently followed throughout the development stage of this thesis was to satisfy an optimum trade-off between computational complexity and system performance. This was made feasible by achieving the minimum required performance objectives (e.g., data-rate, BER), while considering the implementation cost of the proposed processing architecture. Hence, each PHY-layer algorithm was optimized accounting for its real-time FPGA-implementation cost and its impact on the overall system performance. Crucial factors to enable the latter are listed below:

  - Minimized latency due to the stringent timing requirements, inherent of real-time broadband signal processing. Highly-parallel pipelined processing structures are required, which need to be designed at algorithm level (e.g., data dependencies, bounded-complexity mathematical formulation).
  - Maximized precision exploiting the full dynamic range, the goal of which is to provide individual signal bit-length that satisfies an optimum trade-off between the number of fixed-point bits and the resulting numerical precision, accounting for the complete system functional interdependences and expected signal conditions (e.g., in a pilot-based channel estimation a low SNR distorts the estimated channel even if infinite precision is utilized for the DSP).
  - The previous two items require to be combined with a highly optimized internal memory structure and an intelligent control-plane ensuring the precise synchronous operation of each block composing the considered system (see Section 3.4 for a more detailed description).

- **FPGA-based implementation accounting for the DFE:** the DFE is a critical component affecting the operation and performance of the entire mobile receiver, especially when considering high mobility channel conditions. As already mentioned in this chapter, due to the challenging implementation of the DFE when developing a real-time PHY-layer, many authors avoid to implement it, which basically implies that the PHY-layer misses a very influential processing stage with a definitive role in real-life receivers. Hence, in contrast to the mentioned researchers, the DFE has been designed to serve the real-time requirements of the case studies mentioned in Chapters 5 and 6, whilst it is neither assumed a perfectly synchronized nor an error-free signal. Indeed the implemented DFE comprises a full synchronization mechanism, featuring as well the AGC and digital down-conversion stages. Moreover, its design is accounting for realistic signal impairments. The inclusion of the DFE is having the effects listed hereafter:

  - The complexity of the baseband design is greatly increased, since additional bit-intensive DSP is required.
  - The global performance of the system is critically affected by the operation of the DFE. This is due to the elevated precision required for its internal calculations, but also due to the need to provide a precise synchronous control of the data-flow forwarded to the remaining baseband blocks (e.g., gain-adjustment accounting for the utilized frame-format and timings of the internal processing architecture).
  - The DFE is the physical interface with the analog front-end. This means that its design requires to account for the hardware specifications of the target development platform (e.g., bit-width and interpolating capacities of the ADC/DAC stages, physically available gain-adjustment circuitry, dedicated I/O latencies).

- **High-performance PHY-layer solutions based on real-world technology:** a wide signal bandwidth of 20 MHz (combined with a 2048-point FFT) has been considered in accordance to the proposed BWA standards. Furthermore, in an attempt to capture close-to-real-world system specifications, the implemented PHY-layers are relying on the mobile WiMAX and LTE wireless communication standards:

  - In the case of mobile WiMAX, the standard-related operations at baseband are considerably increasing the implementation complexity (e.g., subcarrier permutation, grouping in logical structures and construction of sets of related OFDM symbols; see Chapter 5).
  - The most basic PHY-layer structuring was developed to be compliant with the standard. Then it was subsequently extended by adding advanced (off-standard) communication schemes requiring adaptivity.
  - The channel models utilized to realistically evaluate the resulting real-time prototypes have been selected in accordance to the recommendations given in the standards.

- **Realistic real-time mobile channel evaluation:** on top of the hardware-introduced impairments, frequency selective mobile channel models were used to configure a real-time radio channel emulator, in order to attain the most realistic channel propagation conditions. The provision of real-time mobile channel conditions requires of a complex laboratory setup resulting in additional hardware-introduced impairments, which also need to be considered during the design stage.

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided implementation results	Complete system implementation?	Functional/behavioural verification
[119]	CMOS implementation of a MIMO-OFDM transceiver	4x4 MIMO 20 MHz BW, 256-point FFT WLAN standard	Custom HDL	Silicon metrics with RTL design details	Yes	Emulation in a multi-FPGA platform
[120]	VLSI design of a MIMO sphere decoder for optimized area and throughput	Up to 4x4 MIMO Up to 64-QAM	Custom HDL	Silicon metrics with gate-level design details	No (isolated PHY-layer processing block)	Numerical simulation
[121]	CS-based channel estimation for OFDM-based receivers	SISO 20 MHz BW LTE standard	Custom HDL	Silicon metrics with RTL design details	No (isolated PHY-layer processing block)	Numerical simulation
[122]	SIC-based MIMO detector	4x4 MIMO 16-QAM	Custom HDL	Silicon metrics with RTL design details	No (isolated PHY-layer processing block)	Numerical simulation
[123]	VLSI implementation of a MIMO-OFDM transceiver	4x4 MIMO 80 MHz BW, 512 subcarriers WLAN standard	Custom HDL	Silicon metrics with gate-level design details	Yes	Numerical simulation
[124]	VLSI implementation of an OFDM-based transceiver	SISO 40 MHz BW, 128-point FFT WLAN standard Sensing-based subcarrier allocation	Custom HDL	Silicon metrics with gate-level design details	Yes	Numerical simulation
[125]	ASIC implementation of a high-throughput channel estimator for OFDM receivers	4x4 MIMO 20 MHz BW LTE standard	Custom HDL	Silicon metrics with RTL design details	No (isolated PHY-layer processing block)	MATLAB/C floating-point simulation (Vienna LTE simulators)

(Continued on next page)

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided implementation results	Complete system implementation?	Functional/behavioural verification
[126]	Power-efficient MIMO detector	2x2 MIMO Up to 64-QAM	Custom HDL	Silicon metrics with RTL design details Throughput vs consumption analysis	No (isolated PHY-layer processing block)	Numerical simulation
[127]	MIMO near-MLD with channel preprocessing	8x8 MIMO QPSK	Custom HDL	Silicon metrics with gate-level design details	No (isolated PHY-layer processing block)	Numerical simulation (using both floating- and fixed-point arithmetic)
[128]	Self-configuring FFT/IFFT hardware macrocell for OFDM-based systems	Up to 4x4 MIMO Up to 4096 Multiple OFDM-based standards are considered	Custom HDL	Silicon metrics (considering different technologies and configurations) with gate-level design details	No (isolated PHY-layer processing block)	RTL and gate-level simulation (NC-Sim)
[129]	SVD-based MIMO channel estimator	4x4 MIMO 64-QAM	Custom HDL	Silicon metrics with gate-level design details	No (isolated PHY-layer processing block)	No functional validation is detailed
[130]	Reduced hardware-complexity design of a MIMO sphere decoder	Up to 16x16 MIMO 16 MHz BW	Simulink with System Generator design combined with an ASIC library of bit-true equivalent designs	PAR results and silicon metrics with gate-level design details	No (isolated PHY-layer processing block)	Emulation in a multi-FPGA platform
[131]	Feasibility study of high-performance MIMO-OFDM receivers including adaptivity	2x2 and 4x4 MIMO up to 20 MHz BW AMC, sphere decoding LTE standard	Efficient C-based HLS from low-level optimized C++-code	Synthesis results and silicon metrics with RTL design details	Yes	Numerical simulation

(Continued on next page)

Ref.	Use case	Considered specifications and operating conditions	Utilized design and implementation method	Provided implementation results	Complete system implementation?	Functional/behavioural verification
[132]	Rapid prototyping of a parallel searching-based sphere detector for MIMO-OFDM systems	4x4 MIMO 64-QAM	Efficient C-based HLS from low-level optimized C++-code	Synthesis results and silicon metrics with gate-level design details	No (isolated PHY-layer processing block)	Numerical simulation
[133]	Co-design flow coupling a custom ASIC architecture with the SDR-code development	2x2 MIMO LTE-Advanced receiver (2x20 MHz BW signals)	Low-level optimized C-code combined with ASIC-optimized HLS tools	RTL design details Benchmarking	No (custom DSP-optimized processor)	Numerical simulation (using both floating- and fixed-point arithmetic)
[134]	Custom DSP-RAM processor targeting LST Coded receivers	Efficient software-based implementation of a LST decoder for a 4x4 MIMO system	Custom HDL code	Synthesis results and silicon metrics with gate-level design details Benchmarking vs DSP processors	No (custom DSP-optimized processor)	C++ simulation (using both floating- and fixed-point arithmetic)
[135]	Custom co-processor for MIMO decoders	Efficient software-based implementation of a QR decomposition decoder for a 4x4 MIMO system	Custom HDL code	Synthesis results and silicon metrics with gate-level design details Benchmarking vs DSP processors	No (custom DSP-optimized processor)	Numerical simulation (using fixed-point arithmetic)
[136]	Custom wideband spectrum-sensing processor for cognitive applications	Weak-signal detection in a 64 MHz radio BW with adjacent-band interference	Custom HDL code	Synthesis results and silicon metrics with RTL design details	No (custom DSP-optimized processor)	Numerical simulation Emulation in a multi-FPGA platform

Table 3.9: Synopsis of the encountered literature related to VLSI design.

3.4 Contribution

The contribution of this thesis is focusing on the largest possible subset of the discussed motivation subjects. To satisfy the design and implementation goals it was essential to provide innovating RTL design techniques achieving by this way a performance-efficient FPGA-based PHY-layer realization. A helpful companion towards this end was the complete design, implementation and validation methodology that was applied to the developed case studies.

RTL design principles
A set of advanced RTL-design techniques, that form part of solid FPGA prototyping principles, have been defined in order to achieve a performance-efficient design required for the implementation of high-performance PHY-layer solutions featuring adaptivity. The very same principles can be extrapolated and mapped to the design of other communication systems. A short introduction to the innovating and transferable RTL design principles follows. The items listed hereafter constitute the main axis of the contribution of this thesis around which it has been developed the remaining part of the systems:

- **Adaptive memory structure:** a resource-optimized and minimized-latency memory structure is proposed to cope with the large intermediate storage requirements of the DSP stages encountered at the PHY-layer of modern broadband communication systems. The minimum required set of embedded block RAMs - accounting for the largest possible data-block size at the output of each DSP stage - is grouped as a single adaptive memory entity (i.e., primitive-level HDL coding). This entity enables an easy interfacing and provides a seamless memory access which allows simultaneous write and read operations. Furthermore, it supports the management of subcarrier sets with variable-length, as required by the mobile WiMAX standard. The latter defines various subcarrier-permutation schemes resulting in a flexible frame structure (e.g., coexistence of open and closed-loop communication techniques). Each adaptive memory entity is managed by a complex dedicated controller which enables the transparent synchronous memory access, while accounting for the specific configuration of each received frame. Moreover, the controller is in charge of optimizing the read and write operations by implementing the required scrambling-operations with the minimum possible latency. Additionally, when required, simple baseband operations are applied concurrently with the memory-access (e.g., conjugation of a complex-valued sample). A detailed description of the adaptive memory structures is provided in Section 5.3.4.

- **Resource-reutilization among different communication schemes:** the coexistence of an open and a closed-loop communication scheme and the flexible features of cognitive radio notably increase the demand for baseband processing resources. Hence, when considering limited FPGA processing and memory resources, it is essential to include intelligent resource-reutilization among the different DSP stages. Besides the evident need to identify the similar arithmetic calculations, an accurate RTL design is required to meet the performance needs of each separate configuration (e.g., changes in the frame formatting leading to different latencies or data dependencies). Two indicative examples of the latter are provided in this thesis:

  - A high-performance pipelined structure implementing a 2x2 MIMO STBC-based communication (i.e., open-loop) and its Maximum Ratio Combining (MRC)-based counterpart (i.e., part of a TAS closed-loop scheme) is taking advantage of the similarity of certain arithmetic operations, as detailed in Section 5.3.3. The proposed design achieves a 18% FPGA-resource saving without compromising the performance requirements of the system.
  - In an indoor frequency-reuse scenario, an interference-detection mechanism is implemented by taking advantage of the cross-correlation which provides the basis of the synchronization stage of an LTE-based receiver. The proposed design is detailed in Section 6.3.2.

- **Optimized design for DSP-block saving:** the development of the considered PHY-layers involved bit-intensive arithmetic computations, consuming the majority (if not all) of the available DSP-blocks embedded in the FPGA devices of the target baseband prototyping platform. Hence, a careful design is needed to optimize the utilization of these specialized processing elements as detailed hereafter:

  - Various DSP operations which are utilized to calculate the location of the subcarriers within each OFDM-symbol (i.e., part of the permutation schemes applied in the mobile WiMAX PHY-layer), have been designed at gate-level to achieve an efficient implementation which avoids the utilization of DSP-blocks. Indicative examples are the LUT-based implementation of a modulus or a division by a constant, as detailed in Section 5.3.5.
  - The synchronization stage of a mobile WiMAX receiver features an algorithm optimization that specifically targets a minimized arithmetic-implementation cost in terms of DSP-slices. Hence, a simplified mathematical formulation targeting a minimal use of DSP-blocks is combined with an efficient pipelined RTL architecture design. The complete details of this design are provided in Section 5.3.1.
  - The utilization of complex Finite Impulse Response (FIR) filters is required as part of the interference-detection mechanism utilized in the LTE-based receiver described in Section 6.3.1. The latter is demanding a large amount of DSP-blocks and general purpose FPGA-slices. Hence, a design relying on time-multiplexing is utilized to achieve re-utilization of resources.

- **Speed-optimized pipelined-arithmetic design:** a clear example of architecture optimization, focusing on the provision of a minimized-latency design, that enables the real-time MIMO channel estimation of a high-performance mobile WiMAX receiver is presented in Section 5.3.2. Its pipelined architecture features an in-block dedicated memory structure aimed at the simultaneous retrieval of operands as required by the supported subcarrier permutation schemes.

- **Resource and latency-aware centralized control entity:** when a large number of bit-intensive concurrent operations are sequentially executed in a real-time system, a fine-grain timing control of the diverse synchronous data-path stages is required. Moreover, when run-time adaptive DSP is required or other advanced processing structures are utilized (e.g., resource-re-utilization among different communication schemes or adaptive memory structures), it is essential the existence of a centralized control unit in charge of configuring each group of operations comprising a logical processing stage (or block), according to the current frame structure (e.g., adaptive subcarrier allocation or coexistence of various subcarrier permutation schemes). This control unit could also provide the interfacing with the MAC-layer, though it has not been developed for none of the two cases studies. Two indicative examples of centralized control are presented hereafter:

  - The coexistence of the STBC and the TAS mechanisms in a mobile WiMAX receiver (each based on a different subcarrier permutation scheme) requires a precise centralized control, on top of the already mentioned adaptive memory structure and resource-reusing techniques. Indicatively, FPGA-logic reuse principles are applied to the baseband processing blocks of the receiver which utilizes both schemes in different time instants. The previous is made possible through a centralized control unit, as described in Section 5.3.6.

  - The interference-management scheme proposed in the LTE-based scenario of the second use case, makes use of a complex state machine that jointly and simultaneously drives the operation of the synchronization and the interference-detection logic. The complete details are provided in Section 6.3.3.

Design, implementation and verification methodology
The design, implementation and on-board testing of high-performance wireless communication systems under realistic conditions implies an undertaking with high stakes. Thus, the adoption of a well-structured design, implementation and verification methodology is a paramount requirement. The employed design flow follows an iterative and modular rationale adding robustness to the innovating digital techniques proposed herein and can be directly re-utilised in similar PHY-layer developments. An overview of the steps describing this methodology is given next:

- An HLPL-based model is utilized as the entry point of the design, accounting for the system specifications. Moreover, the high-level system model plays a crucial role in the algorithm selection stage.

- An off-line testbedding approach is utilized to capture realistic signals, by utilizing the HLPL-model of the transmitter and a laboratory setup (e.g., using a channel emulator and RF equipment). Furthermore, the captured signals are also including the impairments introduced by the analog front-end (e.g., the target FPGAs are utilized as a large buffer to capture the outputs of the ADCs).

- The previous captures are used to refine the HLPL-model (e.g., modifications may be required to the initially selected algorithms).

- Custom HDL code is produced for each logical block of DSP operations utilized in the HLPL system-model. The captured signals are used to optimize the RTL architecture (e.g., dynamic range adjustment) and to verify the HDL code (e.g., precision comparison versus its HLPL-based counterpart).

- The baseband design is integrated onto the target platform (e.g., interfacing with the ADC/DAC stages and other relevant I/O).

- A thorough laboratory-debugging stage is used to refine the RTL design. The configuration of the testing stage starts from a from a baseband-to-baseband cabled connection and incrementally adds more hardware equipment such as RF front-ends and a channel emulator

- Once a stable prototype is obtained, an exhaustive measurement campaign is put in place. The captured data is post-processed using HLPL software tools in order to extract the required performance metrics (e.g., averaging large number of captured repetitions for a various range of signal conditions).

Details of the described design methodology are given in Chapter 4.

Produced publications
The following publications have been produced throughput the development of this thesis:

1. *Design, Implementation and Testing of a Real-Time Mobile WiMAX Testbed Featuring MIMO Technology* [Font-Bach 10]
Conference paper published in the Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), volume 46: 'Revised Selected Papers of the International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities (TridentCom)', May 2010.

2. *A Real-Time FPGA-Based Implementation of a High-Performance MIMO-OFDM Mobile WiMAX Transmitter* [Font-Bach 11a]
Conference paper published in the LNICST, volume 81: 'Revised Selected Papers of the International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT)', May 2011.

3. *A Real-Time FPGA-based mobile WiMAX Transceiver Supporting Multi-Antenna Configurations* [Bartzoudis 11]
Conference paper published in the Proceedings of the Argentine Conference on Micro-Nanoelectronics, Technology, and Applications (CAMTA), August 2011.

4. *Processing-Demanding Physical Layer Systems Featuring Single Or Multi-Antenna Schemes* [Font-Bach 11c]
Conference paper published in the Proceedings of the European Signal Processing Conference (EUSIPCO), September 2011.

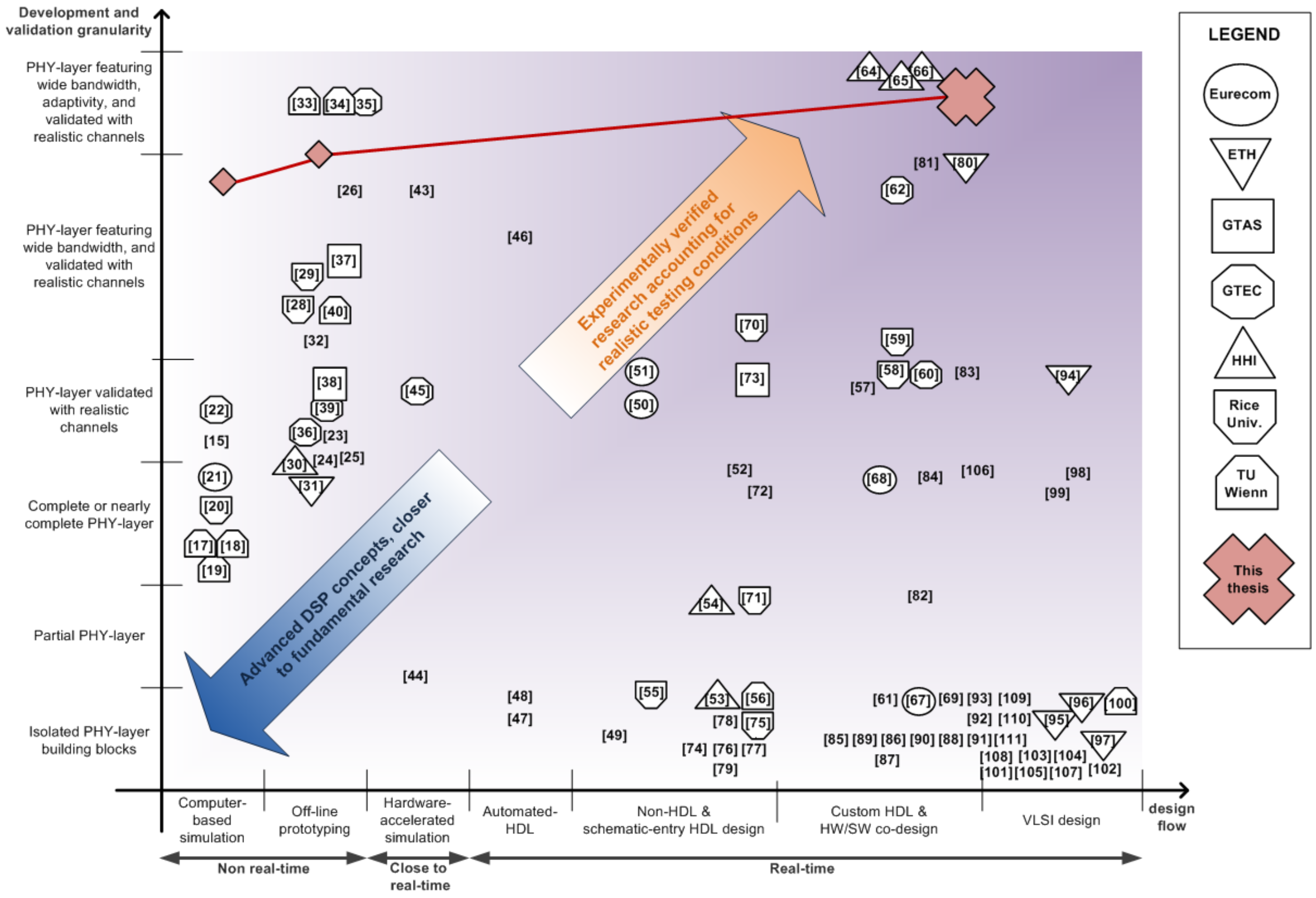


Figure 3.1: Graphical synopsis of the reviewed literature and contributions of this thesis.

5. *A Real-Time MIMO-OFDM mobile WiMAX Receiver: Architecture, Design and FPGA Implementation* [Font-Bach 11b]
   Journal paper published in ELSEVIER Computer Networks, volume 55, number 16: 'Special Issue on mobile WiMAX', November 2011.

6. *A Real-Time FPGA-based Implementation of a High-Performance MIMO-OFDM Transceiver featuring a Closed-Loop Communication Scheme* [Font-Bach 12a]
   Conference paper published in the Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), October 2012.

7. *MATLAB as a Design and Verification Tool for the Hardware Prototyping of Wireless Communication Systems* [Font-Bach 12b]
   Chapter in 'MATLAB, A Fundamental Tool for Scientific Computing and Engineering Applications, Volume 2' published by InTech, 2012.

8. *Hardware-efficient implementation of a Femtocell/Macrocell interference-mitigation technique for high-performance LTE-based systems* [Font-Bach 13b]
   Conference paper (accepted to be) published in the Proceedings of the International Conference on Field Programmable Logic and Applications (FPL), September 2013.

The following publication is about to be submitted:

9. *An experimental real-time implementation of an interference management scheme in a LTE-based Macrocell-Femtocell HetNet scenario* [Font-Bach 13a]
   Journal paper to be submitted to EURASIP Journal on Wireless Communications and Networking in June 2013.

Synopsis of the provided contribution

In an attempt to provide an objective conclusion to this chapter, Figure 3.1 provides a graphical synopsis of the relation of this thesis with the reviewed literature, as well as that of its contribution.

Chapter 4

Design, implementation and verification methodology

The aim of this chapter is to offer an insight into the development flow utilized for the PHY-layer prototyping goals that have been set for this thesis. A well-structured design, implementation and verification methodology provides the backbone upon which the system has been developed. It is important to underline that one of the contributions of this thesis is encountered in the structure and reusability of this methodology in similar experimentally driven PHY-layer developments.

4.1 Considered development framework

On top of the development conditions detailed in Section 3.3, the proposed development framework also accounts for the following:

- **Extended HDL design flow:** in order to attain a performance-efficient design it is required to carefully define the RTL architecture of each DSP component (i.e., optimum trade-off between its computational complexity and the resulting numerical precision, given a specific target FPGA device). While the proposed custom HDL design and implementation approach allows the designer to control all the important aspects of the designed architecture, it must feature three fundamental characteristics to achieve the desired design efficiency:

    1. *Modularity:* a modular hierarchical design of the PHY-layer is essential to enable the substitution, modification or extension of individual DSP blocks. Modularity facilitates as well the testing and debugging stages (e.g., fine-grain verification, co-simulation of intermediate processing stages at different levels of abstraction).
    2. *IP and design re-utilization:* the utilization of HDL-based DSP libraries improves the whole implementation process in terms of development time, by providing an error-free and efficient digital realization of complex baseband operations. A number of fundamental DSP functions are available as pre-verified IP cores (e.g., FFT, Coordinate Rotation Digital Computer - CORDIC, fixed-point dividers).

3. *Necessity for a low-level design:* even the most powerful modern FPGA devices are offering a (large but) limited number of processing and storage resources. Hence, when considering the real-time prototyping of high-performance wireless communication systems featuring adaptivity, an enormous data-transfer, storage and processing capacity is required at baseband. Thus, an efficient RTL design might not always be sufficient. In such cases, designers resort to gate-level design techniques which are injected in the RTL architecture of the system under development (e.g., directly instantiate FPGA-primitives in the HDL code, preventing the automatic logic-inference that would be typically performed during the synthesis stage).

- **Accurate configuration of the EDA framework:** it is of uttermost importance to properly configure the software tools supporting the HDL-based implementation (e.g., selection of the most appropriate synthesis and PAR features).

- **Realistic test and validation conditions:** a thorough assessment of the considered systems requires to realistically account for all those signal impairments originated from the testing environment. This for instance requires to account for realistic channel fading, interference and noise test and operating conditions. On top of that, the use of a complicated hardware setup requires a delicate configuration to enable the evaluation of the implemented PHY-layer under the desired realistic and at the same time controllable conditions.

- **Debugging and performance assessment:** the baseband design is finally integrated onto the target FPGA-based development platform. The latter provides the necessary I/O means that facilitate the laboratory debugging stage (e.g., external memories allowing to capture data, test points to visualize the analog signal). Furthermore, the FPGA vendors are providing tools to visualize in real-time the variation of the internal baseband signals (i.e., digital logic-analyser). Both of the mentioned aspects need to be considered at design-time and might result in the utilization of additional FPGA-resources (e.g., embedded RAM blocks). The data captured during the iterative testing and debugging stages needs to be post-processed in order to be utilized in a computer-based simulation or to extract the required performance metrics. For a complete performance assessment of wireless communication systems that feature mobility, a large set of captures needs to be acquired and analysed, which is a non-trivial and lengthy process.

4.1.1 Considered development platform

For the construction of the proposed design, implementation and verification methodology it is required a generic development setup, like the one depicted in Figure 4.1. A brief description of the considered hardware components follows:

- **Baseband prototyping board(s):** a set of processing elements is disposed to implement the designed PHY-layer. Depending on the technology of the target processor, the same processing elements might also provide

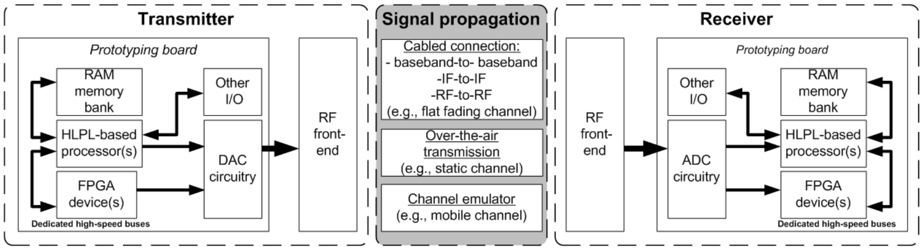


Figure 4.1: Generic development platform.

the required control interfaces (e.g., to program the FPGA devices or to capture data) or to host implementations of the complete network stack.

- **Analog front-end:** at the transmitter the analog front-end provides the required circuitry to synthesize an analog signal, within the designated RF bandwidth, from the baseband samples, whereas at the receiver it produces an IF or a baseband signal (depending on the technology) ready to be processed digitally. The analog front-end comprises the following:

  1. *RF front-end:* the main tasks carried out at the RF front-end are the low-noise amplification, the downconversion from RF to IF, or upconversion from IF to RF, and the suppression of out-of-band unwanted signals (e.g., analog filtering to eliminate noise and spurs).
  2. *ADC/DAC circuitry:* the ADC devices sample the analog IF signal resulting, which results in replicas of the signal's spectrum, whose formation depends on the utilized bandwidth and IF center frequency. As a result a digital signal is provided to the baseband. A Digital Down Converter (DDC) is typically used then to produce the I/Q data from the sampled IF signal. The DAC devices synthesize an IF signal from the I/Q baseband samples, by utilizing interpolating filters.

- **Signal propagation:** depending on the verification goals the following three signal propagation scenarios could be considered:

  1. *Cabled connection:* a cabled baseband-to-baseband, IF-to-IF or RF-to-RF connection provides a flat-fading channel and is important to be used during the preliminary testing and debugging stages in order to characterize the hardware-introduced impairments.
  2. *Over-the-air transmission:* the utilization of antennas provides real-world signal propagation conditions. The over-the-air transmissions usually focus on validating specific test scenarios (e.g., indoors environments or line of sight with quasi-static signal propagation conditions). This is mainly due to the fact that field testing and measurement campaigns have a high logistic, operating and administrative cost (e.g., renting an RF band of interest), especially when mobility is required (e.g., a large number of repetitions are required for each experiment).
  3. *Channel emulator:* the utilization of a RF channel emulator allows to consider a broad range of channel conditions, including mobility,

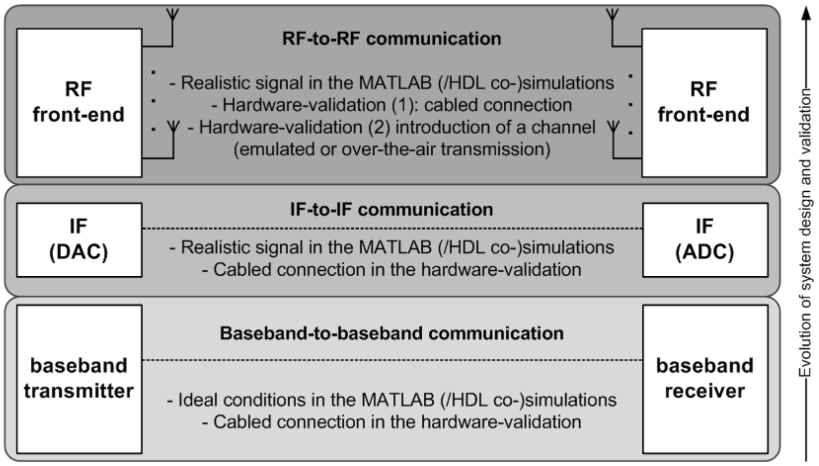

RF-to-RF communication
RF front-end
- Realistic signal in the MATLAB (/HDL co-)simulations
- Hardware-validation (1): cabled connection
- Hardware-validation (2) introduction of a channel (emulated or over-the-air transmission)
RF front-end
IF-to-IF communication
IF (DAC)
- Realistic signal in the MATLAB (/HDL co-)simulations
- Cabled connection in the hardware-validation
IF (ADC)
Baseband-to-baseband communication
baseband transmitter
- Ideal conditions in the MATLAB (/HDL co-)simulations
- Cabled connection in the hardware-validation
baseband receiver
Evolution of system design and validation

Figure 4.2: Multi-stage testing strategy.

from the commodity of a laboratory-based setup. However, such equipment is an expensive asset to acquire or rent. Unavoidably an emulator introduces additional hardware impairments to the RF signal (which need to be measured, characterized and compensated in order to provide a well-conditioned signal to the receiver).

Although the presented development platform can be potentially utilized in all the design and implementation approaches discussed in Chapter 2, the considered flow is built around the FPGA technology. Moreover, it is tailored for developments based on a custom RTL architecture (relying on optimized IP cores and low-level optimizations wherever it is required).

4.2 Proposed design, implementation and validation methodology

The development and validation of the PHY-layer of a processing demanding real-time wireless communication system requires a wide range of skills, resources and time. Hence, an incremental design and implementation flow, relying on a multi-stage testing strategy, is the corner stone for the success of the overall development flow, since it provides a robust and flexible framework, that favours design re-use and serves the goals of a modular and hierarchical flow.

4.2.1 Multi-stage testing strategy

A fundamental guideline that applies throughout the design, implementation and on-board validation phases is the need for an incremental testing strategy, as described by Figure 4.2. A layered testing approach allows the step-by-step characterization of the system.

The multi-stage testing strategy starts from a baseband-to-baseband system testing under ideal conditions (e.g., no impairments are accounted, flat channel response is assumed). This is made feasible by developing and simulating first the high-level behavioural model of the system (e.g., in MATLAB) and then, based on that, develop and simulate the equivalent RTL representation. In order to achieve similar testing and validation conditions in the target hardware platform, a direct (cable) connection of the transmitter and receiver baseband implementations is required.

When the behavioural validation of the system is accomplished, the testing stage is extended to include the signal conversion stages (i.e., ADC and DAC). It implies the extension (and re-simulation) of the HLPL and HDL code to include the processing blocks that provide the interfacing to the conversion circuitry (e.g., interpolating filters, DDC). The next step is to validate the FPGA implementation in real-time using an IF-to-IF cabled-connection (i.e., connecting via a cable the output of the DAC device with the input of the corresponding ADC device). During this on-board functional verification it is essential to extract data captures, in order to optimize the operation of the processing blocks providing directly interfaced to the signal conversion stages (i.e., through additional simulations; e.g., to adjust the digital filtering stage of the DDC, accounting for any undesired out-of-band component that might be introduced by the baseband processing boards). Additionally, specialized signal analysis equipment are normally assisting the laboratory validation of the produced signal (e.g., spectrum analyser).

The final testing stage can be divided into two sub-stages; the first includes a direct cable connection of the RF front-ends, which results in a signal that accounts for the specific hardware impairments of the target testbed. The extraction of signal captures at the receiver baseband (e.g., post-ADC) permits the full characterization of such impairments. Hence, additional computer-based simulations are necessary for the different models of the system accounting for the realistic hardware-impaired signal. Finally, the desired channel conditions must be included to the considered scenario, either by using antennas or a real-time radio channel emulator. This results in a signal that accounts for both realistic channel conditions and hardware-introduced impairments. Yet again, signal-captures and re-simulations are required to fully evaluate the system (which might result in further debugging of the developed models).

4.2.2 Incremental development flow

Figure 4.2 shows the design, implementation and verification methodology. A commonly accepted starting point for any incremental development is the design of a baseline version of the target system, featuring downscaled specifications (e.g., when targeting a MIMO PHY-layer featuring AMC, a single-antenna open-loop communication scheme might constitute its baseline design). Indeed, the initial design-efforts should focus on the core signal processing algorithms, including as well the identification of the most critical aspects of the overall system architecture (e.g., operations requiring of high-throughput pipelined structures, dimensioning the required memory structure and control plane).

The end of a major design cycle is reached when the performance of the resulting prototype is finally validated on real-time hardware and no further modifications are required. The proposed methodology explicitly addresses the incremental introduction of more advanced features (i.e., by iterating over the complete development flow). A relative low effort is required to extend the operation and functionalities of the developed prototype; while the first iteration is normally expected to be extremely time consuming, the following ones can be dramatically shortened. This is mainly due to the fact that modular and reusable code is already available (e.g., passing from a single antenna to a multi-antenna design and implementation, many DSP blocks can be either directly reused or easily extended), while at the same time the critical parts of the design and the system bottlenecks are well defined. The same applies to the hardware platform which is already thoroughly studied and characterized.

The text that follows describes in detail the different stages of the employed design, implementation and verification methodology.

Stage I: Develop a high-level model of the transmitter

The first vital requirement for the design of any wireless communication system is the definition of the transmitted signal. Based on this signal model it is made possible to start the development of a HLPL-based model of the transmitter (e.g., using the MATLAB framework). At this stage, the considered test scenario plays a key role, because it defines the target application, the environmental conditions and the performance requirements. In most of the cases the modelling of the transmitted signal is bound to the specifications of a wireless communication standard, which defines for instance parameters such as the duplexing mode, the format and length of the frame, the number, value and location of the pilot tones, the guard-band size, the inter-carrier spacing and available bandwidth and also the FFT size. Furthermore, it details the operating RF bands and channel models to be used for the test and validation of the developed systems.

At this initial development stage the high-level model is not taking into account a number of practical implementation conditions and parameters; indeed the modelling makes use of floating-point arithmetic, assumes unlimited processing resources during design/simulation time and does not account for signal-impairments (e.g., channel effects, noise or hardware-introduced non-idealities). This allows the system designer to concentrate on the definition of the basic DSP architecture, facilitating likewise the behavioural simulation of the system.

Stage II: Hardware-validation of the baseband transmitter model

The HLPL model of the transmitter can be rapidly tested and hardware-validated by using off-line prototyping approach. This is made feasible by interfacing the MATLAB model with Commercial-Off-The-Shelf (COTS) RF instrumentation, signal generation equipment and signal conversion boards. More specifically, the output I/Q vectors of the baseband transmitter model can be directly loaded to a VSG[1] which, with the help of an arbitrary waveform generator, provides in real-time the I/Q signals. The VSG features DAC and RF upconversion circuitry able to provide a real-world signal at the selected RF band. At this stage, it can be also verified the compliance of the resulting RF signal with the selected wireless communication standard by using third party software tools (e.g., Agilent Vector Signal Analyser - VSA).

The inclusion of additional signal impairments (e.g., real-time emulation of a selected channel, addition of noise or of CFO) should not be considered at this stage (i.e., such operating conditions make unreliable the capturing of test-vectors until the DFE of the receiver is developed and tested at the target FPGA

[1] Many VSGs provide the means to produce RF signals directly from MATLAB [VSG].

board). Therefore, a cabled RF-to-RF connection should be used to introduce the transmitted signal to the down-conversion stage of the receiver. After being digitized at the acquisition boards, the resulting samples can be captured by using the FPGA as an intermediate buffer. The captured data constitute realistic test vectors that can be either used for developing the equivalent model of the receiver or for testing, debugging or improving the transmitter model.

Stage III: Develop a high-level model of the receiver
The next step is the modelling of the signal processing algorithms of the receiver. As in the case of the transmitter, the ideal high-level model of the receiver uses floating-point logic and does not have any design limitations in terms of processing and memory resources. Therefore, the functional testing of the complete system is conducted by running a software simulation of the transmitter and receiver models (i.e., ideal baseband-to-baseband signal).

The simulations can also make use of the test vectors captured in the previous step. Although the design is still not constrained by the limitations of the entire hardware processing platform, the performance of different algorithms can be rapidly investigated, including those whose computational requirements makes their real-time prototyping challenging (i.e., estimation of the maximum theoretical system performance). Finally, this stage allows the testing of different candidate algorithms targeting a specific processing stage (e.g., channel estimation). Likewise, it is possible to achieve a coarse grain assessment of the computational cost of different algorithms and their impact on the overall system performance. This greatly facilitates the selection of the algorithms that satisfy a trade-off between complexity and performance.

Stage IV: Signal impairment modelling
At this point, it is available the ideal HLPL model of the entire system and it should be gradually adapted according to those real-world impairments introduced by the hardware setup and also those defined in the target test scenario. This process is inherently incremental; in the first iteration, only those impairments originated by the intrinsic features of the target hardware components can be realistically introduced (i.e., by using the data-captures obtained at Stage II). Indicative signal impairments stemming from the operation of the underlying hardware equipment are the ADC/DAC quantization-noise, the coupling of the transmitted signal with that of the LO and the introduction of a DC level by the baseband processing boards.In the absence of a DFE operating in real-time at the baseband processing boards (used to develop the real-time receiver), the effects of the signal propagation channel are based on computer-generated static channel models. Other effects of the signal due to the physical propagation medium, such as the noise, are emulated using pseudo-random software routines. Once the DFE is implemented, tested and validated, baseband data can be captured in its output using the complete hardware setup of the testbed (Stages X and XI). This data help to verify the remaining of the HLPL-based design of the receiver. In fact, it is likely that certain DSP algorithms will have to be modified or slightly adapted to account for more realistic signal propagation conditions (e.g., real-time mobile channel emulation, inclusion of CFO, presence of interference, inclusion of Additive White Gaussian Noise - AWGN - generators).

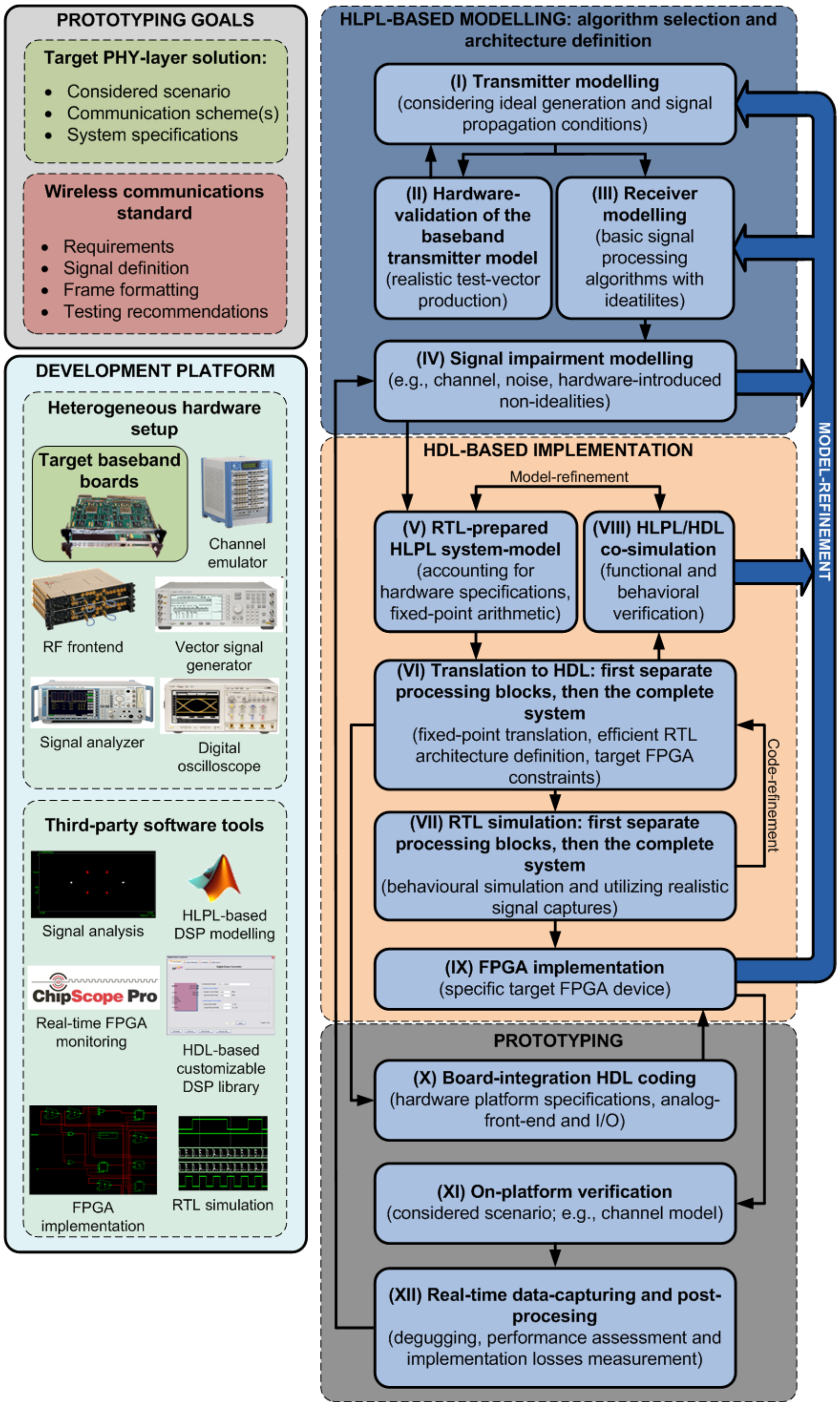

PROTOTYPING GOALS
Target PHY-layer solution:
Considered scenario
Communication scheme(s)
System specifications
Wireless communications standard
Requirements
Signal definition
Frame formatting
Testing recommendations
DEVELOPMENT PLATFORM
Heterogeneous hardware setup
Target baseband boards
Channel emulator
RF frontend
Vector signal generator
Signal analyzer
Digital oscilloscope
Third-party software tools
Signal analysis
HLPL-based DSP modelling
ChipScope Pro
Real-time FPGA monitoring
HDL-based customizable DSP library
FPGA implementation
RTL simulation
HLPL-BASED MODELLING: algorithm selection and architecture definition
(I) Transmitter modelling (considering ideal generation and signal propagation conditions)
(II) Hardware-validation of the baseband transmitter model (realistic test-vector production)
(III) Receiver modelling (basic signal processing algorithms with ideatilites)
(IV) Signal impairment modelling (e.g., channel, noise, hardware-introduced non-idealities)
HDL-BASED IMPLEMENTATION
Model-refinement
(V) RTL-prepared HLPL system-model (accounting for hardware specifications, fixed-point arithmetic)
(VIII) HLPL/HDL co-simulation (functional and behavioral verification)
(VI) Translation to HDL: first separate processing blocks, then the complete system (fixed-point translation, efficient RTL architecture definition, target FPGA constraints)
(VII) RTL simulation: first separate processing blocks, then the complete system (behavioural simulation and utilizing realistic signal captures)
Code-refinement
(IX) FPGA implementation (specific target FPGA device)
MODEL-REFINEMENT
PROTOTYPING
(X) Board-integration HDL coding (hardware platform specifications, analog-front-end and I/O)
(XI) On-platform verification (considered scenario; e.g., channel model)
(XII) Real-time data-capturing and post-procesing (degugging, performance assessment and implementation losses measurement)

Figure 4.3: Overview of the design, implementation and verification methodology.

Stage V: RTL-prepared HLPL system-model

In order to have a HLPL baseband model of the system that provides a close match to the prerequisites of RTL coding, further modifications and refinements have to be conducted. The MATLAB models of the transmitter and receiver have to account for the hardware platform specifications (i.e., ADC/DAC features, internal buses, I/Os, available FPGA-resources, etc.) and for the translation to fixed-point arithmetic. Furthermore, it is widely known that not all MATLAB structures or functions are implementable in a FPGA. Even if equivalent HDL constructs exist, they are used during simulation time but do not serve for logic synthesis (e.g., a for-loop construct with undefined number of iterations). Moreover, MATLAB includes several pre-compiled DSP functions (e.g., FFT) and provides abstract arithmetic operators (i.e., the user calls the same operator independently of the type of the operands). For instance, the '*' operator provides the multiplication for integer, real or complex numbers, arrays and matrices. Although these MATLAB features provide a powerful workbench for users, it is common quite a mistake to underestimate the computational complexity and the internal arithmetic calculations of such operations, especially when they are meant to be mapped on a real-time RTL-based implementation. The importance of this evaluation stage for the mapping of the MATLAB model to RTL code is crucial and may result in selecting different algorithms and lightweight versions of pre-compiled arithmetic functions. Another important task is to estimate the storage and intercommunication needs. This is made feasible by including in the MATLAB model a high-level representation of the memory and control planes.

Stage VI: Translation to HDL

Each processing block that forms part of the RTL-prepared HLPL system model needs to be translated to HDL code. In this thesis, it has been employed a custom HDL coding strategy which enables the performance-efficient implementation of modern wireless communication systems. Hand-written HDL coding requires an optimized RTL design that relies on the principles described in Section 3.4.

Every logical grouping of arithmetic functions or baseband processing stages (initially defined in the high-level MATLAB model) are implemented in an incremental fashion. The goal of the RTL design is the optimum utilization of the processing resources of the specific FPGA devices that are available in the target hardware testing platform. At this stage, the HDL interfacing with the baseband prototyping boards is apparently omitted (this is gradually introduced in Stage X, when preliminary on-platform verification efforts are conducted).

Stage VII: RTL simulation

A RTL behavioural simulation is first conducted for each of the designed HDL processing blocks. Specialized software tools were used for this reason (i.e., Mentor Graphics ModelSim). The simulation of independent processing blocks allows a reasonable simulation-time (more details are provided in Stage VIII). Moreover, the RTL simulation of the complete system (or large parts of it) can be kept for the final validation stage.

The RTL simulations, accounted for both synthetic stimulus and realistic test vectors. The first, though, should only be used for functional verification purposes. When it is required to thoroughly evaluate the performance of the designed RTL architecture, it is crucial to utilize a realistically conditioned input. This in various stages of the development cycle resulted in modifications of the baseband algorithmic (e.g., a more complicated channel estimation technique is required), and thus a repetition of Stage I/III.

Stage VIII: HLPL/HDL co-simulation

The co-simulation of the HDL and HLPL descriptions of each DSP stage completes the computer-based analysis of the designed system. The contribution of these co-simulations to the overall development flow is two-fold: first, it allows to verify the behavioural match of both models (i.e., the same response is produced to a given input) and, more importantly, it allows to quantify the losses that are introduced by the fixed-point RTL representation (i.e., compared to the high-level model). In fact, transforming the floating-point arithmetic of the high-level model to fixed-point one is a key procedures for adapting a computer-based simulation of a system to a real-time implementation targeting a FPGA device. During this procedure the HDL design and its RTL-ready HLPL counterpart are refined iteratively, until it is found the optimum quantization of all the arithmetic calculations. In other words, the fixed-point representation (i.e., definition of the precise bit-width and decimal point) of the signals is tuned to satisfy a trade-off between precision and computational complexity (full details are provided in Section 4.3). At the same time, the HDL/HLPL co-simulations play an essential role in the configuration of the IP cores utilized in the RTL architecture.

Additionally, the HLPL/HDL co-simulations are also providing the means to speed up the computer-based simulations of the complete system (or of a large part of it). This for instance may occur when a RTL processing block is combined with the HLPL-based model of the remaining of the system. The simulation of models featuring different abstraction levels is especially useful when the refinement of the basic HLPL-model is required (i.e., loop-back to Stage I/III). This leads in rapidly evaluating new algorithms of a specific baseband processing stage, while the remaining of the RTL design is already validated (i.e., by taking advantage of the low cost of modifying a HLPL-based model).

Stage IX: FPGA implementation

Once the simulated RTL code is functionally stable it is implemented by targeting a physical device with the help of vendor-specific proprietary FPGA software tools (i.e., Xilinx ISE for the FPGA implementations considered herein). This stage has two main objectives:

1. *Estimation of the implementation complexity:* the synthesis of the HDL code provides an estimation of the FPGA resources required by the RTL design (i.e., the synthesis can be realized for every processing block apart or for the entire system). When combining this with Stages VII/VII it can be assessed the computational cost and the performance of the system under development. In this sense, Stages V/VI will have to be repeated if improvements of the RTL design are required; in a worst case scenario, there might be a need to replace an algorithm in order to meet the required performance and computational cost (i.e., loop-back to Stage I/III).

2. *Digital realization:* once the performance requirements are met at simulation time and the synthesis results are satisfactory, the HDL code is fully implemented for a given target FPGA device (i.e., PAR). The generated binary file is the one used to program the FPGA device, enabling likewise the hardware-testing of the designed DSP solution.

Stage X: Board-level HDL code integration

In order to test the implemented system in a chosen FPGA-based board, it is first necessary to interface the developed HDL code with on-chip hard-wired components and on-board peripherals, buses, controllers, memories, ICs, analog components (i.e., digitally controlled) and custom/standard I/Os. This stage is usually denoted as the on-board HDL code integration. In COTS-based FPGA boards, the manufacturers usually facilitate this process by providing an HDL FPGA firmware, that includes all or most of the required HDL interfaces with the on-chip and on-board components. Some very common examples of this integration phase include the interfacing with PGAs, RAMs, DACs and gigabit transceivers. Finally, a common testing and debugging practice is also to instantiate control and monitoring cores within the developed HDL code, which with the help of proprietary software can monitor the internal digital signals in real-time (e.g., Xilinx ChipScope Pro). The inclusion of this extra logic has to be carefully considered since it might occupy a great portion of the on-FPGA memory blocks (and in some specific cases of generic FPGA slices).

Stage XI: On-platform verification

This stage includes the functional verification and debugging of the developed system using a full real-time hardware platform. Each time that a new (or modified) design needs to be experimentally validated, it is required to follow the incremental testing approach described in Section 4.2.1.

During this phase, a heterogeneous equipment setup (usually denoted as real-time testbed) is necessary to reproduce the scenario under consideration and provide the close-to-real-world testing conditions. The use and configuration of such testbeds usually imply a non-negligible setup cycle. Additionally, it is also vital the proper utilization of a set of third-party software tools and APIs in order to enable the communication with the test and monitoring points mentioned at Stage X.

Stage XII: Data capturing and post-processing

An important outcome of the on-laboratory verification procedure and, thus, of the whole development flow, is the extraction of experimental data, which can be utilized for the purposes described hereafter::

1. *Debugging-purposes:* the on-FPGA logic analyzers have a limited capacity in monitoring and capturing data; this is either due to the software controlling them or due to the limited embedded memory blocks available at the target FPGA device (considering as well, that a great part of them will be consumed by the custom baseband design). For this reason, when the implemented system is not operating at all, or when it is under-performing (e.g., due to precision problems), it is required to capture big chunks of data at certain parts of the baseband system. This data recording is typically done by using an external RAM with a reasonable size. In order to

achieve this is required to include a priori an HDL interface with the RAM controller, which will allow to tap the signals or ports of interest. Retrieving this data allows to return in previous development stages and debug either the high-level system model or the HDL behavioral simulation.

2. *Performance-assessment:* to thoroughly assess the performance of the developed PHY-layer, it is required to conduct an extensive measurement campaign. This implies testing and capturing data using various channel models (under different mobility conditions), different levels of noise and/or interference. If mobility is considered a large number of repetitions are required, for a certain channel model using different noise levels, in order to provide a meaningful performance analysis. More specifically, for each of the selected noise levels, at least 100 realizations of the selected mobile propagation channel are generated (each one with a different seed) and used to configure in an iterative way, the channel emulator of the testbed. Data is captured at the baseband receiver (at different points) during each one of these testing-iterations (i.e., if 100 channel models are generated per noise level, then the testing iteration will have to be 100 as well). The post-processing of this data, eventually allows to calculate typical PHY-layer performance metrics, by averaging the data captures for each of the defined noise levels. In certain testing scenarios it is required to consider different velocities or channel models together with a number of other signal impairments (e.g., SIR and CFO conditions).

4.2.3 Benefits of the proposed methodology

While at a first glance it might seem that the proposed incremental methodology results in a long development cycle, a closer look at the presented development framework justifies its structure; this is mainly due to the fact that the efficient real-time PHY-layer implementation of high performance communication systems requires a robust, modular and iterative design, implementation and test approach, which is subject to realistic signal conditions and real-world hardware constraints. More specifically, the benefits of the presented methodology can be summarized in the following points:

- *Construction of a robust pre-verified FPGA IP library:* the combination of an incremental development flow with the design of a hierarchical modular architecture, not only is beneficial for the modification or extension of a baseline system, but also allows to obtain a library of efficiently designed DSP algorithms.

- *Creation of realistic test vectors:* the experimentally obtained signal captures are not only serving the debugging and performance assessment necessities of the system under development, but also provide realistic inputs to be utilized in a wide range of simulations or other similar developments. For instance, realistic HLPL-based channel models can be constructed from the conducted measurement campaigns, which helps obtaining a more close to real-life performance assessment of algorithms developed at simulation space.

- *Realistic high-level modelling:* each DSP algorithm comprising the HDL-based development, also features its corresponding representation in the

HLPL model. The latter can be used as a test-reference in fundamental-oriented research initiatives that exclusively make use of computer-based simulations.

- *Accelerated hybrid developments:* the developed HLPL-based models can be directly used in an off-line prototype or in a hardware-accelerated simulation.

- *Involvement of a HLPL-based framework in all development-stages:* the utilization of HLPL models and its corresponding working environment (e.g., MATLAB) at all development-flow stages, results beneficial at different levels. First, during design-time the availability of an extensive software library of ready-to-use DSP algorithms allows the rapid evaluation of different architectural solutions. On top of that, MATLAB facilitates the interaction with laboratory equipment, allowing for an enriched design flow.

- *Robust development:* the proposed incremental flow allows the system designer to decompose the development of an extremely complex system onto several stages (i.e., starting from the baseline version and incrementally adding features until the completion of the system). Therefore, it allows to fully concentrate on the efficient design of the underlying DSP architecture, accounting for the specifications of the considered scenario and the target hardware technology. At each iteration the implementation bottlenecks are known better, while realistic data and a library of efficient HDL-HLPL algorithms are ready to be reused (i.e., lower development time, robust and efficient design).

- *Avoidance of unnecessary iterations:* the utilization of the target prototyping platform from the early stages of the design, allows to account for realistic signal conditions, even before defining the RTL architecture of the system. Thus, it can be avoided the need for major modifications of the system design when the HDL coding stage is in an advanced stage (e.g., if the DFE is only simulated accounting for simplified-synthetic signal conditions, then it is likely to present an incorrect behaviour when tested in the hardware platform; this may have a strong impact on the remaining processing stages of the receiver, which could eventually lead to a major redesign of the existing HDL code).

4.3 Development-flow challenges

The proposed methodology defines an adequate roadmap that greatly assists the efficient design, implementation and validation of real-time baseband systems featuring a high bandwidth, adaptive behaviour and accounting for realistic signal conditions. However, a series of challenges need to be properly considered and confronted throughout the development flow. In order to get a better understanding of the magnitude and heterogeneity of the issues that need to be addressed, this section provides an overview of those factors that are crucial for the success of the proposed development flow. Without loss of generality, it can be claimed that the same factors apply in a number of other occasions, where high performance real-time baseband systems need to be efficiently realized and validated with the help of real-life experimental testbeds.

Translation to fixed-point arithmetic

The FPGA-based prototyping of wireless communication systems implies the use of fixed-point logic at baseband. This is a significant design constraint that has to be evaluated considering that HLPL-based modelling (e.g., MATLAB) is based by default on floating point arithmetic. In general terms the floating-point operations dramatically increase the FPGA logic utilization and result in lower clock speeds and longer pipelined structures when compared to fixed-point logic. The designers are responsible for mapping the HLPL algorithms to an HDL-based fixed-point logic, which in fact is a demanding and non-trivial task. The latter implies that all internal processing stages of the transmitter and receiver (both in MATLAB-space and HDL-design space) have to be appropriately simulated to tune them at an optimum fixed-point dynamic range, applying numerous truncation and scaling steps to achieve the best arithmetic precision. Additionally, each of the implemented HDL blocks has to be co-tested with the equivalent portion of the floating-point HLPL model to ensure that the system performance is not compromised. A very handy modification of the MATLAB model that assists the comparison with the equivalent RTL code is to apply quantization at the outputs of selected processing blocks that represent functional partitions of the design. This quantization process emulates the fixed-point logic. Thus, a recursive process is required to discover the optimal achieved performance of the designed system, which is conditioned by the processing and memory resources of the target FPGA device, the additional inherent constraints of the hardware platform and the minimum required yield of the system (e.g., BER, average data rate).

Signal impairments

The real-life equipment comprising experimental wireless communication testbeds feature signal processing elements that are subject to impairments and performance degradation. Certain impairments are testbed-specific, since the analog components and ICs comprising the boards and instruments feature different performance and specifications for each testing platform.

The baseband implementation at the transmitter features degradation due to use of fixed-point arithmetic and also due to performance limitations of the digital filters. The signal impairments at the DAC device, apart from the performance of its embedded interpolation filters, could also be generated from various other sources (e.g., losses due to Peak-to-Average Power Ratio - PAPR - backoff, quantization noise, harmonic distortion, DC offset, jitter/timing errors, DAC roll-off tilt). The analog front-end of the transmitter comprises mixers, gain amplifiers, LO, frequency synthesizers and Power Amplifiers - PAs - that might feature a number of impairments (e.g., amplitude and phase imbalance, LO phase noise, intermodulation and cross-modulation distortion, spurs, distortion due to analog filtering, PA non-linearities, thermal noise).

The signal impairments at the receiver's analog front-end are due to similar functional anomalies to the ones identified before for the transmitter (e.g., operation of Low-Noise Amplifiers - LNAs, band definition filters, duplexers, mixer, LO, frequency generation and gain stages); the same applies for the ADC devices. On top of this the digital baseband part at the receiver needs to estimate and compensate a series of signal impairments that are created as part of the regular operation of the boards and instruments comprising the entire testbed. Along with distortion and other effects added by the channel, the baseband signal at the receiver might feature high-levels of CFO, Sample Frequency Offset (SFO), phase noise, DC-offset, I/Q gain and phase mismatch and other non-linearities. These signal degradations and impairments are critical for the operation of the receiver.

Channel and mobility effects

Unless the transmitter and receiver are placed within an echo-free environment, the transmission is normally subject to multi-path propagation. If the transmitter or receiver moves, the channel and its fading will be time varying. Mobility in a wireless transmission causes several different effects; the most demanding one to confront is the rate at which the channel changes. If the mobility is low (e.g., when the channel can be assumed to be stationary for the duration of a complete symbol or data packet) the channel can be estimated by means of a preamble or similarly known sequence. However, if mobility is high, then the channel changes are significant during a symbol period (i.e., fast fading). Fast fading requires the baseband processor (e.g., DSP, FPGA, ASIC) to track and recalculate the channel estimation during reception of user payload data. This in very high mobility speeds implies that the baseband processor must timely trace the channel changes during the reception of a data slot, which severely challenges the baseband design timings and performance. Additionally, the latter creates frequency offsets, which however can be relatively easy estimated and compensated at the receiver. The number and distribution of pilots in the transmitted signal (different for each wireless communication standard) also has key importance for the receiver's performance (i.e., channel estimation).

FPGA design partitioning

The ever increasing demand for bandwidth and the inclusion of complex PHY-layer algorithms, imposes the simultaneous use of numerous baseband processing devices to serve the run-time processing load. In the case of FPGAs, the integration of the HDL code with the on-board peripherals and the design partitioning in various FPGA devices creates additional challenges. For instance, both the embedded or inter-board buses need to accommodate the communication of a big volume of data at a high throughput. Moreover, the FPGA partitioning requires among other design-measures, a well thought functional separation of the baseband processing blocks and the development of a custom communication protocol for the intra-board (FPGA-to-FPGA) and board-to-board communication. This protocol has to guarantee data integrity and provide error correction capabilities if necessary. Although a number of modern ultra-high capacity FPGA devices offer the required horse power to develop the PHY-layer of processing demanding systems (i.e., the FPGA partitioning complications do not apply), it is naturally expected that the need for more bandwidth together with the constant inclusion of intelligent signal processing features at baseband will bring back in the near future the multi-device baseband processing quest.

Design and implementation software tools
The performance of the EDA software depends on the underlying hardware architecture and OS of the computer system that hosts them, the programming style of the designer and the signal processing complexity of the application under development. More specifically, when FPGA EDA tools are used to design, simulate and implement wideband communication systems, a series of challenges has to be addressed. For instance the HDL simulations suffer long simulation times (which results in a prolonged debugging stage), the FPGA implementation tools produce giga-byte-sized intermediate files, timing is hard to meet (especially in dense FPGA implementations), multiple clock domains raise metastability issues and the overall implementation strategy varies according to the target device. Moreover, while 64-bit versions of modern FPGA EDA software tools are available nowadays, only a fraction of them takes advantage of the capacity of modern multi-core microprocessors for reducing the overall simulation or implementation time. Finally, the previously mentioned EDA tools feature OS-related limitations and deadlocks when excessive amounts of RAM memory are required.

Testbed characterization
Setting up and configuring a real-time wireless communication testbed, that focuses on PHY-layer algorithm prototyping, implies the separate and joint testing of the diverse type of instruments and signal processing boards that form part of it. This allows researchers to achieve the desirable operating conditions and performance, according to the specifications of a certain wireless communication standard. In order to accomplish this task, certain performance features of the equipment comprising the testbed need to be characterized and analysed, a fact that helps to identify performance bottlenecks at the beginning of the development (e.g., the impact of the DAC and ADC resolution on the precision of critical baseband processing algorithms and the limitations imposed of the processing and memory capacity at baseband).

Stability
Real-time testbeds are commonly having performance stability issues due to the interdependencies among instruments in the signal processing path (baseband, ADC/DAC, RF conversion and channel); it is quite common, for example, that the power-level of the signal between cabled instruments or hardware boards may unexpectedly variate out of the defined range due to some sort of failure, mismatch or loose connection of cables, adapters, attenuators, on-board connectors or other components. This may significantly change the overall expected performance of the developed baseband transmitter and receiver, making also hard the traceability of the problem during system-debugging.

4.4 Description of the GEDOMIS® testbed

The developments presented in the thesis have been hosted in the GEDOMIS® (GEneric hardware DemOnstrator for MIMO Systems) testbed [GED], an experimental platform that comprises a complete set of high performance baseband prototyping boards (FPGA and DSP-based), signal generation equipment, high-

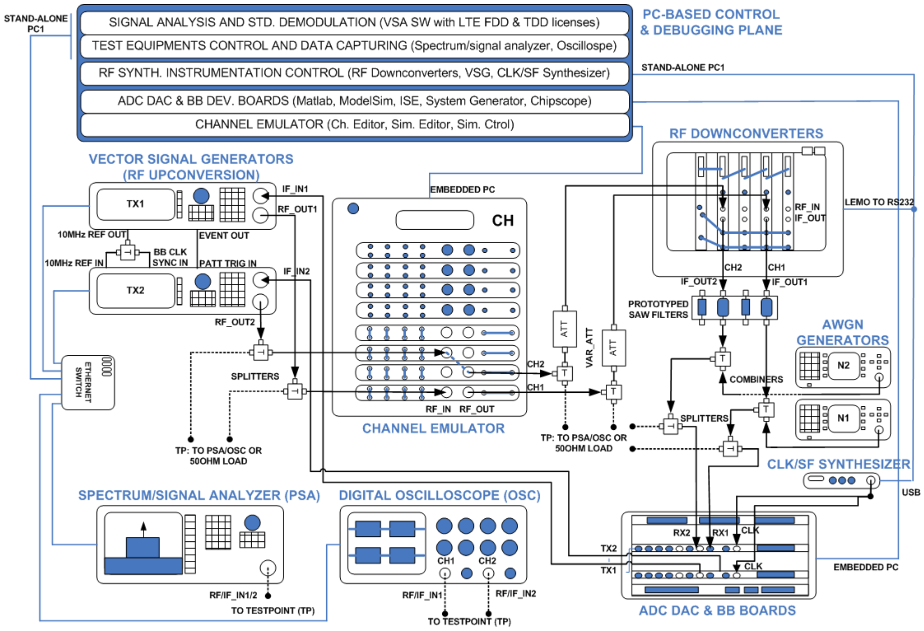

STAND-ALONE PC1
SIGNAL ANALYSIS AND STD. DEMODULATION (VSA SW with LTE FDD & TDD licenses)
TEST EQUIPMENTS CONTROL AND DATA CAPTURING (Spectrum/signal analyzer, Oscillospe)
RF SYNTH. INSTRUMENTATION CONTROL (RF Downconverters, VSG, CLK/SF Synthesizer)
ADC DAC & BB DEV. BOARDS (Matlab, ModelSim, ISE, System Generator, Chipscope)
CHANNEL EMULATOR (Ch. Editor, Sim. Editor, Sim. Ctrol)
PC-BASED CONTROL & DEBUGGING PLANE
STAND-ALONE PC1
VECTOR SIGNAL GENERATORS (RF UPCONVERSION)
TX1
TX2
IF_IN1
RF_OUT1
10MHz REF OUT
10MHz REF IN
EVENT OUT
BB CLK SYNC IN
PATT TRIG IN
IF_IN2
RF_OUT2
ETHERNET SWITCH
SPLITTERS
TP: TO PSA/OSC OR 50OHM LOAD
EMBEDDED PC
CH
CH2
CH1
RF_IN
RF_OUT
CHANNEL EMULATOR
ATT
VAR_ATT
ATT
TP: TO PSA/OSC OR 50OHM LOAD
RF DOWNCONVERTERS
RF_IN
IF_OUT
LEMO TO RS232
CH2
CH1
IF_OUT2
IF_OUT1
PROTOTYPED SAW FILTERS
AWGN GENERATORS
N2
N1
COMBINERS
SPLITTERS
CLK/SF SYNTHESIZER
USB
SPECTRUM/SIGNAL ANALYZER (PSA)
RF/IF_IN1/2
TO TESTPOINT (TP)
DIGITAL OSCILLOSCOPE (OSC)
CH1
CH2
RF/IF_IN1
RF/IF_IN2
TO TESTPOINT (TP)
RX2
RX1
CLK
TX2
TX1
CLK
EMBEDDED PC
ADC DAC & BB BOARDS

Figure 4.4: Utilized GEDOMIS testbed setup.

end RF front-ends, real-time channel emulation, signal analysis instruments, specialized software tools and APIs.

A schematic representation of GEDOMIS is shown in Figure 4.4, which includes the majority of the hardware equipment and boards used in the default set-up of the testbed (i.e., real-time point-to-point DL communication). The equipment of GEDOMIS can be divided in three major categories: the RF front-ends, the channel emulation and the baseband processing elements. Testing and debugging instruments are facilitating the validation of the developed system at different levels (baseband and RF). A brief description of the high-end instruments and boards used in each of the mentioned categories is given in the following sections.

4.4.1 RF front-ends

The RF-upconversion stage of the GEDOMIS testbed is implemented by the *Agilent ESG4438C VSG* (Figure 4.5a), which combines outstanding RF performance and sophisticated baseband generation to deliver calibrated test signals at baseband, IF, and RF frequencies up to 6 GHz. The ESG4438C VSG offers an internal baseband generator with arbitrary waveform and real-time I/Q capabilities, ample waveform playback and storage memory, and a wide RF modulation bandwidth (i.e., 80 MHz). By using two or more of these signal generators MIMO transmission is enabled; currently GEDOMIS is hosting two of these VSGs.

The signal generation capabilities of the ESG4438C VSG are augmented when it is combined with the *Agilent Signal Studio* software suite, which is enabling the rapid and flexible creation of application-specific test signals, based on specific wireless communication standards (e.g., IEEE 802.11, IEEE 802.16

(a) Upconversion: Agilent ESG4438C VSG.

(b) Downconversion: MCS Echoteck Series RF 3000T Tuners.

Figure 4.5: High-end instrumentation comprising the RF section of the GEDOMIS testbed.

or 3GPP LTE), at baseband, RF, and microwave frequencies. Indicatively, the Signal Studio Toolkit is used to download custom I/Q waveforms to the VSG, which allows to rapidly deploy an off-line transmitter (i.e., the arbitrary waveform generator plays back the I/Q baseband vectors of the MATLAB transmitter model).

For the RF-downconversion it is utilized the *Mercury Computer Systems (MCS) Echotek Series RF 3000T Tuners* (Figure 4.5b), which is a reconfigurable high-performance 4-channel platform to down-convert general wireless standard signals from RF to IF. The equipment must be placed within the frame of a complete RF, data acquisition and signal processing platform and performs high dynamic range, wide bandwidth, excellent phase noise, single or multi-channel phase-coherent operation and ultra fast tuning speed. The basic 3000T tuner is comprised of two single-slot VME modules: the 3000RF receiver module and the 3000S synthesizer module (using Direct Digital Synthesizer - DDS - technology), which act as an 20 MHz to 3 GHz RF-to-IF downconverter. The tuner provides both an IF at 140 MHz (65MHz bandwidth) and a baseband output suitable for direct input into an external ADC. For multi-channel applications, additional 3000RF modules can be integrated with the 3000S synthesizer module to form a multi-channel fast-tuning radio solution; currently GEDOMIS features four RF modules. The system can run in a master/slave configuration, such that the system can be dynamically changed to run in either independently tunable or phase-coherent operation. This allows the system to be cabled up once and reconfigured on the fly via the proprietary control software.

Table 4.1 summarizes the most relevant specifications of the above detailed high-end RF equipment.

4.4.2 Channel emulation

The real-time channel emulation capacity of GEDOMIS facilitates the verification and testing of system designs prior to field-trials. The real-time multichannel emulation is provided by the *Elektrobit (EB) Propsim C8 Channel Emulator* (Figure 4.6a), which is a technology-independent radio channel emulator supporting all major wireless communication standards and signal types in a

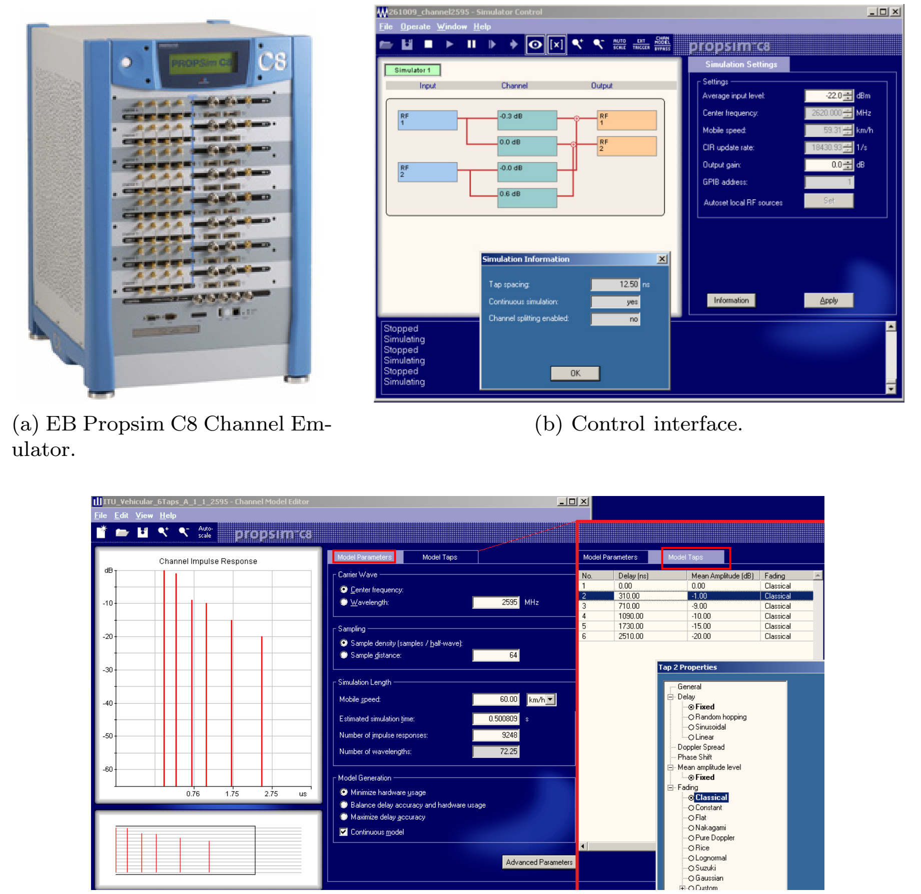
Equipment	Main specifications
Agilent ESG4438C VSG	250 kHz to 6 GHz 80 MHz bandwidth +17 dBm output power <-134 dBc phase noise at 20 kHz offset ±1 ppm internal reference accuracy
MCS Echoteck Series RF 3000T Tuners	20 MHz to 3 GHz 65 MHz bandwidth Manual gain control 85 dB <-115 dBc phase noise at 10 kHz offset ±0.5 ppm internal reference accuracy

Table 4.1: Specifications of the RF section of GEDOMIS.

broad frequency range (350MHz to 6GHz, featuring a 70 MHz bandwidth) covering established and future technologies. It allows users to perform realistic and accurate radio channel emulation supporting the development of most demanding wireless applications (e.g., beamforming, MIMO or SDR); the current configuration of the channel emulator allows the deployment of a single 4x4 MIMO communication (or two 2x2 MIMO channels). The physical radio channel characteristics, such as frequency, multipath propagation, fast fading, dynamic delays, attenuation, noise, interference and shadowing, can be emulated independently on each channel. The user can utilize all major standardized channel models, create its own radio channel models (i.e., up to 48 taps; e.g., from MATLAB) or utilize measured channel data. Furthermore, the channel emulator is provided with application software that facilitates significantly its configuration. First, a channel model editor allows to define the statistical radio channel specific parameters that characterize the (time variant) channel impulse response. It creates a simplified statistical channel model for a situation where the mobile station is moving away from the base station at a constant speed, by creating and editing tap files and setting other relevant parameters such as the operation center frequency, the mobile speed and the channel emulator resource usage (Figure 4.6c). Additionally, a simulation editor is used to create and edit the simulation block layout (i.e., define the connections between inputs, channels models used and outputs). Finally, a simulator control interface (Figure 4.6b) allows to load and emulate the generated channel models. Moreover, it can be used to change the simulation parameters (e.g., additional RF output gains on top of those defined by the channel model) and control the playback of the channel emulation (e.g., play, stop or bypass).

Besides the channel emulator, the testbed is equipped with two *Applied Instruments (AI) NS-3 RF Noise Source* devices. These broadband RF noise generators provide an extremely flat AWGN signal from 5 to 2.15 GHz. The output level adjusts in 0.1 dB steps over a 30 dB range and can be ON/OFF pulse modulated. It is mainly used to facilitate the assessment the system performance under variable SNR conditions (i.e., addition of AWGN at IF for BER versus SNR testing).

Table 4.2 summarizes the most relevant specifications of the above detailed channel emulation and AWGN generation equipment.



Settings

Average input level:	-22.0	dBm
Center frequency:	2620.000	MHz
Mobile speed:	59.31	km/h
CIR update rate:	18430.93	1/s
Output gain:	0.0	dB
GPIB address:	1	
Autoset local RF sources	Set	

Simulation Information

Tap spacing:	12.50	ns
Continuous simulation:	yes	
Channel splitting enabled:	no	

Stopped
Simulating
Stopped
Simulating
Stopped
Simulating

Carrier Wave
● Center frequency:
○ Wavelength: 2595 MHz

Sampling
● Sample density (samples / half-wave):
○ Sample distance: 64

Simulation Length
Mobile speed: 60.00 km/h
Estimated simulation time: 0.500809 s
Number of impulse responses: 9248
Number of wavelengths: 72.25

Model Generation
● Minimize hardware usage
○ Balance delay accuracy and hardware usage
○ Maximize delay accuracy
☑ Continuous model

No.	Delay (ns)	Mean Amplitude (dB)	Fading
1	0.00	0.00	Classical
2	310.00	-1.00	Classical
3	710.00	-9.00	Classical
4	1090.00	-10.00	Classical
5	1730.00	-15.00	Classical
6	2510.00	-20.00	Classical

General
Delay
  ● Fixed
  ○ Random hopping
  ○ Sinusoidal
  ○ Linear
Doppler Spread
Phase Shift
Mean amplitude level
  ● Fixed
Fading
  ● Classical
  ○ Constant
  ○ Flat
  ○ Nakagami
  ○ Pure Doppler
  ○ Rice
  ○ Lognormal
  ○ Suzuki
  ○ Gaussian
  ○ Custom

(a) EB Propsim C8 Channel Emulator.

(b) Control interface.

(c) Channel model editor.

Figure 4.6: The main channel emulation GUI.

4.4.3 Baseband processing elements

The heart of the GEDOMIS testbed is provided by the *Lyrtech Advanced Development Platform (ADP)*, which comprises signal conversion and baseband processing boards; the current board-configuration utilized in GEDOMIS is shown in Figure 4.7. All the boards boards comprising the ADP are installed in a chassis, which provides a compact Peripheral Component Interconnect (cPCI) backplane interconnection between them. One of these boards is a general purpose computer that runs the proprietary ADP software, including a dedicated Graphical User Interfaces (GUIs) for controlling and interacting with each board that is connected to the cPCI bus. Considering that all described boards fitted in the ADP are making use of the same the cPCI bus, the role of the PC is not limited to off-line control functions, but it is also thought to be used for various real-time operations such as waveform record and play-back or controlling user-programmable registers in the FPGA at run-time.

The VHS-DAC (Figure 4.8b) and VHS-ADC (Figure 4.8a) signal conversion boards of the ADP are fitted respectively with 4 dual Texas Instruments (TI) DAC 5687 chips, providing up to 8 transmitting channels, and 4 dual ADC 6645

Equipment	Main specifications
EB Propsim C8 Channel Emulator	350 MHz to 6 GHz 70 MHz bandwidth Up to 48 fading paths per channel Propagation delay up to 6.4 ms Mobile speed up to 40,000 km/h
AI NS-3 RF Noise Source	5 MHz to 2.15 GHz 30 dB range with 0.1 dB steps -90 dBm/Hz maximum output power ±2.0 dB flatness over full operating range

Table 4.2: Specifications of the equipment utilized by GEDOMIS for the provision of realistic channel conditions.

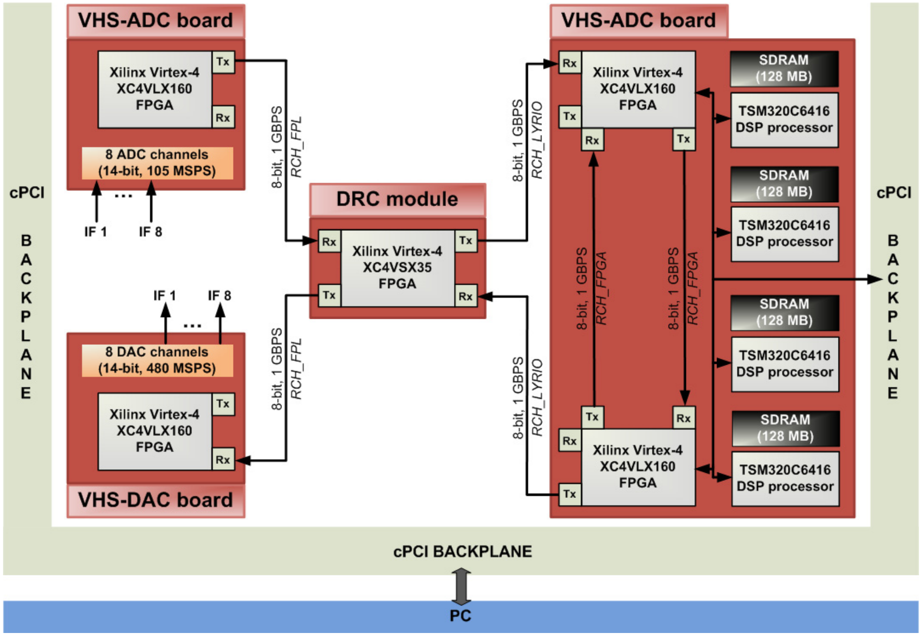
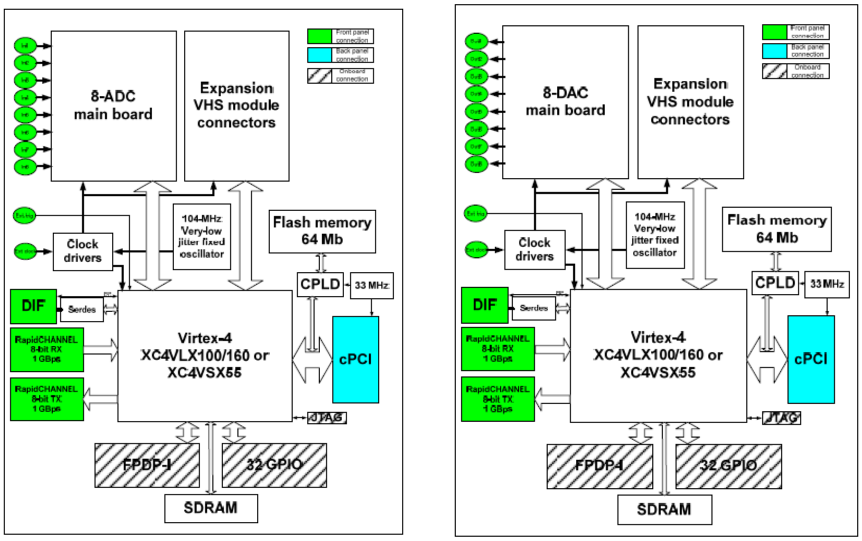

VHS-ADC board
Xilinx Virtex-4 XC4VLX160 FPGA
Tx
Rx
8 ADC channels (14-bit, 105 MSPS)
IF 1
IF 8
cPCI
BACKPLANE
8-bit, 1 GBPS
RCH_FPL
DRC module
Xilinx Virtex-4 XC4VSX35 FPGA
IF 1
IF 8
8 DAC channels (14-bit, 480 MSPS)
Xilinx Virtex-4 XC4VLX160 FPGA
VHS-DAC board
8-bit, 1 GBPS
RCH_LYRIO
VHS-ADC board
Xilinx Virtex-4 XC4VLX160 FPGA
SDRAM (128 MB)
TSM320C6416 DSP processor
8-bit, 1 GBPS
RCH_FPGA
cPCI
BACKPLANE
cPCI BACKPLANE
PC

Figure 4.7: The basebands boards comprising GEDOMIS.

chips, providing up to 8 receiving channels. These two boards allow the implementation of MIMO systems and other antenna array configurations (e.g, used in radar or terrestrial-to-satellite communications). Each of the eight phase-coherent channels of the DAC and ADC boards shares a common clock source and includes a Linear Technology (LT) 5514 ultra-low distortion IF digitally controlled PGA, featuring a gain-control range of 22.5 dB (in 1.5 dB steps). There is also an on-board 104 MHz crystal and external clock and trigger inputs in both boards. The DAC and ADC boards include a single Xilinx Virtex-4 FPGA device (i.e., XC4VLX160), 128 MB of on-board Synchronous Dynamic RAM (SDRAM) and they are connected to the cPCI chassis through a dedicated backplane connection. Finally, a proprietary full-duplex 1 Giga-Byte Per Second (GBPS) bus (i.e., Rapid Channel) allows the VHS-ADC and VHS-DAC boards to connect with the main baseband processing board.

In order to confront the challenges of wideband bit-intensive baseband op-


Front panel connection
Back panel connection
Onboard connection
In1
In2
In3
In4
In5
In6
In7
In8
8-ADC main board
Expansion VHS module connectors
Ext trig
Ext clock
Clock drivers
104-MHz Very-low jitter fixed oscillator
Flash memory 64 Mb
CPLD
33 MHz
DIF
Serdes
RapidCHANNEL 8-bit RX 1 GBps
RapidCHANNEL 8-bit TX 1 GBps
Virtex-4 XC4VLX100/160 or XC4VSX55
cPCI
JTAG
FPDP-I
32 GPIO
SDRAM
8-DAC main board
Expansion VHS module connectors
Clock drivers
104-MHz Very-low jitter fixed oscillator
Flash memory 64 Mb
CPLD
33 MHz
DIF
Serdes
RapidCHANNEL 8-bit RX 1 GBps
RapidCHANNEL 8-bit TX 1 GBps
Virtex-4 XC4VLX100/160 or XC4VSX55
cPCI
JTAG
FPDP-I
32 GPIO
SDRAM

(a) VHS-ADC.

(b) VHS-DAC.

Figure 4.8: Block diagram of the VHS-ADC and DAC boards.

erations, the ADP includes another baseband signal processing board, namely the Signal Master Quad (SMQUAD, Figure 4.9), able to address demanding data loads and DSP operations. In the configuration utilized in GEDOMIS, this board includes two Xilinx Virtex-4 FPGA devices (i.e., XC4VLX160) and 4 fixed-point DSP processors from TI (i.e., TMS320C6416 running up to 1GHz). The SMQUAD features 128 MB of SDRAM per FPGA and DSP processor. The inter-FPGA communication is accomplished via a full-duplex 1 GBPS bus and the FPGA-to-DSP communication is accomplished via a 32-bit wide bus. As in the case of the VHS-DAC and VHS-ADC boards the SMQUAD board is similarly connected to the chassis via the cPCI backplane. The SMQUAD features an additional daughter-board (DRC), which is connected to the LYRIO+ site of the SMQUAD board and help the latter to establish full-duplex 1 GBPS connections with the VHS-ADC and VHS-DAC boards, through the on-board Rapid Channel connector. Moreover, the DRC board is fitted with a Xilinx Virtex-4 FPGA device (i.e., XC4VSX35).

The most relevant specifications of the Lyrtech ADP are summarized in Table 4.3.

4.4.4 Complimentary hardware components

Clock synthesis

The current testbed setup utilizes two high-precision, ultra low phase noise, single channel microwave sources, namely the *Holzworth HSC1001A* and *HSM1001A RF Synthesizers* (Figure 4.10), both capable of tuning frequencies up to 1 GHz in 0.001 Hz step resolution. Furthermore, this high end RF synthesizers are highly portable (i.e., small form and can be powered through a custom made dual USB cable). The source can be used as RF synthesizer or as baseband system clock or ADC sampling frequency generator, providing both low jitter and

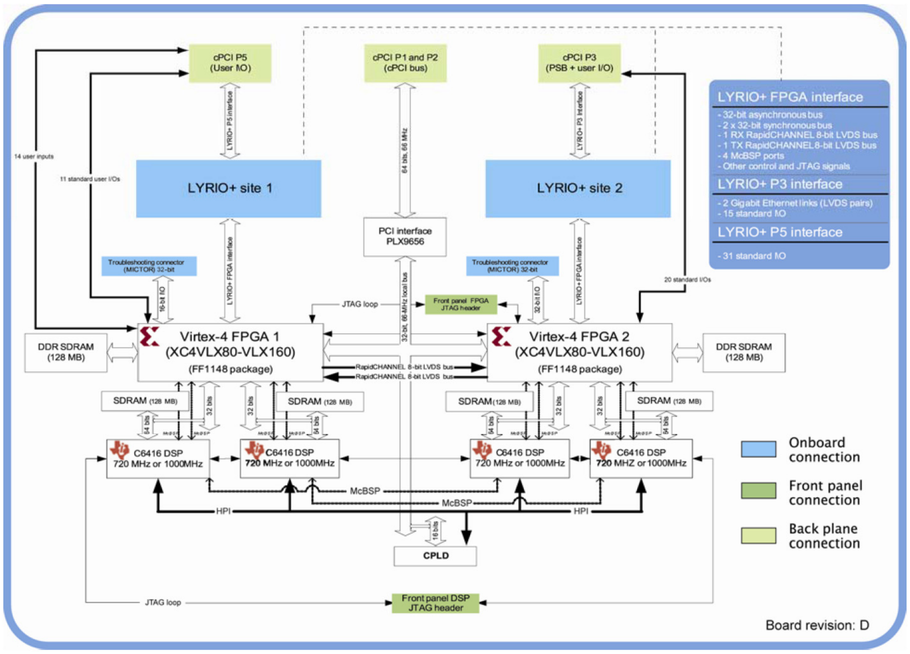


Figure 4.9: Block diagram of the SMQUAD-4 board.

(a) HSC1001A.

(b) HSM1001A.

Figure 4.10: Holzworth microwave source generators.

low phase noise (e.g., which is required for high-density multi-carrier OFDM systems).

An indispensable feature of both microwave sources is their ability to be configured in cascade; i.e., the HSM1001A RF synthesizer can operate according to an input reference signal, which is provided by the HSC1001A module. As a result, true phase and frequency coherent clock signals are produced (e.g., where the period of both signals is related, as typically required in multi-clock domain systems).

Table 4.4 summarizes the most relevant specifications of the above detailed baseband clock signal generation equipment.

Signal analysis

A *Rohde & Schwarz (R&S) FSQ Signal Analyzer* (Figure 4.11a), which offers signal analysis, up to 26.5 GHz, at a demodulation bandwidth of up to 120

VHS-ADC	VHS-DAC	SMQUAD-4	DRC
ADCs: 8xAD6645 (105 MSPS, 14-bit resolution) **Analog inputs:** 0.2-200 MHz BW -18 to 4 dBm **Sampling clock:** Onboard 104 MHz External source **Control & pre-processing:** XC4VLX160 (152064 logic cells, 1056 kb RAM, 96 DSP-slices) **Storage:** 128 MB SDRAM **Off-board I/O:** RapidCHANNEL (1 GBPS, full-duplex)	**ADCs:** 4xDAC5687 (480 MSPS, 14-bit resoultion) **Analog inputs:** 0.3-240 MHz BW -21 to 2.9 dBm **Sampling clock:** Onboard 104 MHz External source **Control & pre-processing:** XC4VLX160 (152064 logic cells, 1056 kb RAM, 96 DSP-slices) **Storage:** 128 MB SDRAM **Off-board I/O:** RapidCHANNEL (1 GBPS, full-duplex)	**FPGA devices:** 2xXC4VLX160 (152064 logic cells, 1056 kb RAM, 96 DSP-slices) **DSP processors:** 4xTMS320C6416 (1 GHz, fixed-point) **Storage:** 6x128 MB SDRAM **Off-board I/O:** RapidCHANNEL (1 GBPS, full-duplex) **On-board Inter-FPGA bus:** LYRIO (1 GBPS) **Backplane connections:** Direct FPGA I/O Direct LYRIO I/O Gigabit Ethernet cPCI interface	**FPGA device:** XC4VSX35 (34560 logic cells, 240 kb RAM, 192 DSP-slices) **Off-board I/O:** RapidCHANNEL (1 GBPS, full-duplex) **On-board Inter-FPGA bus:** LYRIO (1 GBPS)

Table 4.3: Main specifications of the ADP platform.

MHz with the dynamic range of a high-end spectrum analyzer. The equipment has application firmware for digital demodulation measurements for the most common wireless communication standards (e.g., IEEE 802.11 or IEEE 802.16), and for noise figure and gain analysis.

GEDOMIS is also comprising an ultra-high performance *Agilent DSO80804B Infiniium Oscilloscope* (Figure 4.11b). The available scopes support four analog channels with up to 10GHz bandwidth with up to 40 GSPS sample rate. This instrument can be used as multiple-channel analog front-end to extract data for further processing with specialized software (and thus enabling MIMO reception).

Finally, the *Agilent 89600 VSA software* provides superior general-purpose and standard-specific signal evaluation and troubleshooting tools. These tools can be used to dig into the signal and gather the data needed to successfully troubleshoot PHY-layer signal problems. The full set of libraries is available through the educational license of VSA, which enables the required standard-compliance tests (Figure 4.11c).

Table 4.5 summarizes the most indicative specifications of the above detailed signal analysis instrumentation.

4.4.5 Signal impairments resulting from the utilization of GEDOMIS

GEDOMIS as any other similar testbed, features certain signals impairments which have to be accounted on top of the noise and the channel effects. These impairments are due to the specifications and performance characteristics of

Equipment	Main specifications
HSC1001A RF synthesizer	8 MHz to 1 GHz 0.001 Hz resolution -110 to +15 dBm output power range <-131 dBc phase noise at 10 kHz offset Internal reference 100 MHz ±1 ppb internal reference accuracy
HSM1001A RF synthesizer	250 kHz to 1 GHz 0.001 Hz resolution -70 to +10 dBm output power range <-133 dBc phase noise at 10 kHz offset Internal reference 100 Mhz ±1 ppb internal reference accuracy External reference input 10/100 MHz

Table 4.4: Specifications of the baseband clock-generation equipment utilized in GEDOMIS.

Equipment	Main specifications
R& S FSQ Signal Analyzer	20 Hz to 26.5 GHz 120 MHz bandwidth -173 dBm displayed average noise level 235 MSa I/Q memory <-133 dBc phase noise at 10 kHz offset
Agilent DSO80804B Infiniium Oscilloscope	4 analog channels 10 GHz bandwidth 40 GSaPS Noise floor 294 μV (5 mV/div)

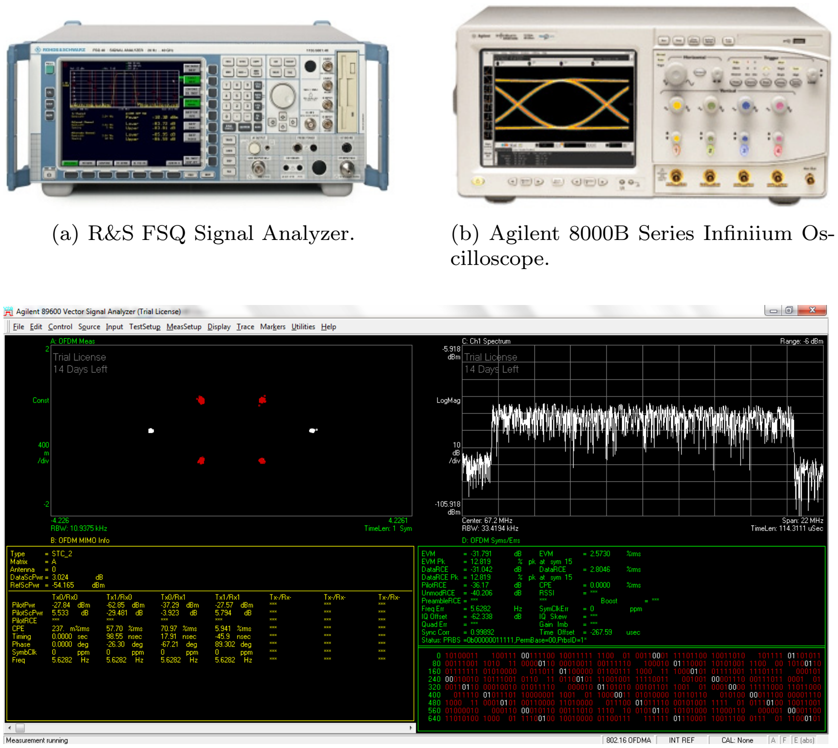
Table 4.5: Specifications of the signal analysis equipment of GEDOMIS.

the equipment comprising the testbed (e.g., channel emulator, RF front-end, baseband signal processing boards, etc.). If the received subcarriers lose their orthogonality due to the mentioned impairments, the performance of the MIMO-OFDM system degrades dramatically. Thus, such signal-impairments have to be determined (i.e., they could vary in different testbeds) and compensated before making the symbol decisions. Hence the signal model that traditionally includes the effects of the channel and the noise has to be contemplated with the testbed specific signal impairments, which have been measured and characterized prior to the validation of the developed system.

The specifications of certain equipment comprising the GEDOMIS testbed contribute in the creation of certain signal impairments, while the specification of others renders common impairments negligible, as detailed hereafter:

(a) High-end RF signal conversion equipment are utilized at the two ends (upconverters in the transmitter and downcoverters in the receiver). This results in I/Q gain and phase imbalance which can be safely ignored especially when considering the RTL implementation of the DFE at the receiver (i.e., certain parts of the designed logic are cyclically-reset in order to compensate accumulated errors). CFO can be synthetically introduced at the baseband transmitter or the RF upconverters. The receiver is then able to estimate and compensate at run-time the introduced CFO.

(b) High precision clock signals are generated at both ends of the point-to-point system. Thus, they can be safely ignored the effects of inaccuracy



```
Type      = STC_2
Matrix    = A
Antenna   = 0
DataScPwr = 3.024     dB
RefScPwr  = -54.165   dBm

           Tx0/Rx0        Tx1/Rx0        Tx0/Rx1        Tx1/Rx1        Tx-/Rx-   Tx-/Rx-   Tx-/Rx-
PilotPwr   -27.84  dBm    -62.85  dBm    -37.29  dBm    -27.57  dBm    ***       ***       ***
PilotScPwr 5.533   dB     -29.481 dB     -3.923  dB     5.794   dB     ***       ***       ***
PilotRCE   ***            ***            ***            ***            ***       ***       ***
CPE        237.  m%rms    57.70   %rms   70.97   %rms   5.941   %rms   ***       ***       ***
Timing     0.0000  sec    98.55   nsec   17.91   nsec   -45.9   nsec   ***       ***       ***
Phase      0.0000  deg    -26.30  deg    -67.21  deg    89.302  deg    ***       ***       ***
SymbClk    0       ppm    0       ppm    0       ppm    0       ppm    ***       ***       ***
Freq       5.6282  Hz     5.6282  Hz     5.6282  Hz     5.6282  Hz     ***       ***       ***
```

```
EVM          = -31.791   dB     EVM         = 2.5730    %rms
EVM Pk       = 12.819    %  pk  at  sym 15
DataRCE      = -31.042   dB     DataRCE     = 2.8046    %rms
DataRCE Pk   = 12.819    %  pk  at  sym 15
PilotRCE     = -36.17    dB     CPE         = 0.0000    %rms
UnmodRCE     = -40.206   dB     RSSI        = ***
PreambleRCE = ***               ***              Boost       = ***
Freq Err     = 5.6282    Hz     SymClkErr   = 0          ppm
IQ Offset    = -62.338   dB     IQ  Skew    = ***
Quad Err     = ***              Gain  Imb   = ***
Sync Corr    = 0.99892          Time  Offset = -267.59   usec
Status: PRBS =0b00000011111,PermBase=00,PrbsID=1*
```

```
  0 10100011   100111 00111100 10011111 1100  01 00110001 11101100 10011010   101111 01101011
 80 00111001 1010  11 00000110 00010011 00111110   100010 01110001 10101001 1100  00 10100110
160 01111111 01010000   011011 01100000 01100111 1000  11 10000101 01111001 11101111   000101
240 00010010 10111001 0110  11 01100101 11001001 11110011   001001 00001110 00111011 0001  01
320 00110110 00010010 01011110   000010 01101010 00101101 1001  01 00010000 11111000 11011000
400   011110 01011101 10000001 1001  01 10000011 01010000 10110110   010100 00011100 00001110
480 1000  11 00010101 00110000 11010000   011100 01011110 00101001 1111  01 01110100 10011001
560 01000010   000110 00010110 00111010 1110  10 01010110 10101000 11000110   000001 00001100
640 11010100 1000  01 11100100 10010000 01100111   111111 01110001 10011100 0111  01 11000101
```

(a) R&S FSQ Signal Analyzer.

(b) Agilent 8000B Series Infiniium Oscilloscope.

(c) Agilent 89600 VSA software.

Figure 4.11: Signal analysis stage of the GEDOMIS testbed.

between the sampling clocks of the transmitter and receiver (in respect to the exact sampling frequency), LO drifts and random phase noise due to LO instability. Moreover, the inherent LO coupling at the transmitter has to be accounted.

(d) The utilization of the AWGN generators allows to precisely control the level of SNR experienced by the receiver.

(e) The chassis housing the signal conversion and baseband processing boards introduces a DC level in the digitalized signal, which has to be accounted in the signal model.

(f) The radio channel emulator allows the reproduction of a wide range of channel conditions, the effects of which are obviously taken into account in the received signal model.

4.4.6 Utilization of GEDOMIS

The instruments or components of the testbed must be appropriately configured, connected or tuned to meet the desirable operating conditions and performance prerequisites (e.g., according to the specifications of a certain standard). GEDOMIS does not offer a unified GUI towards that end. On the contrary, the different equipment, boards and software can be controlled and configured by accessing them as indicated below:

- Different web-based GUIs running on a browser of the ADP computer provide access to the:
  - Agilent E4438C VSGs.
  - R&S FSQ Signal Analyzer.
  - Agilent 8000B Series Infiniium Oscilloscope.
- GUIs running at the computer PC provide APIs for the:
  - MCS Echotek Series RF 3000 4-channel Downconverters.
  - Holzworth HSC1001A and HSM1001A RF Synthesizers.
  - Lyrtech VHS-ADC, VHS-DAC and SMQUAD boards of the ADP.
- The AI NS-3 RF Noise Source can exclusively be programmed in-situ (i.e.,no GUI is available).
- A separate computer (i.e., embedded in the instrument) runs a software that provides access to the EB C8 Channel Emulator.
- A separate computer hosts the Agilent VSA software (i.e., the ADP PC does not meet the performance specs).

In the remaining of the section it is generally described the basic procedure to develop both and off-line testbed and a real-time prototype by using GEDOMIS.

Off-line testbed configuration
As already mentioned before, off-line testbeds typically combine MATLAB simulations at the two baseband ends (i.e., transmitter and receiver) with instrumentation for the signal conversion (i.e., DACs, RF up-conversion, channel emulator or antennas, RF downconversion and ADCs). The basic off-line PHY-layer realization flow utilizing GEDOMIS, as shown in Figure 4.12, is detailed below:

- Utilize the MATLAB model of the transmitter to generate the baseband I/Q output vectors (and store them in a .MAT file).
- Use the Agilent Signal Studio Toolkit to upload the I/Q vectors to the Agilent ESG4438C VSGs.
- Use the ESG4438C VSGs to produce real RF signals:
  - For the generation of MIMO signals, certain adjustments need to be applied. Indicatively, for the two-antenna case, the two ESG4438C are used in a master-slave configuration, where the master baseband generator triggers the slave one by using a waveform marker at the first sample of the time signal that plays back. Taking into account that the process of marker interpretation and trigger signal generation has a certain delay, time alignment of the two signals is applied using a multi-channel oscilloscope.
- Use the EB Proposim C8 channel emulator to apply a (static) channel:

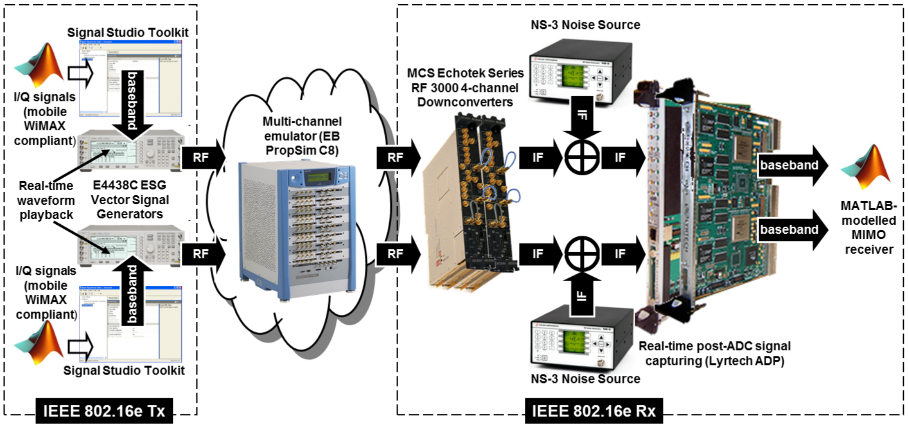

Signal Studio Toolkit
baseband
I/Q signals (mobile WiMAX compliant)
Real-time waveform playback
E4438C ESG Vector Signal Generators
RF
RF
I/Q signals (mobile WiMAX compliant)
baseband
Signal Studio Toolkit
IEEE 802.16e Tx
Multi-channel emulator (EB PropSim C8)
RF
RF
MCS Echotek Series RF 3000 4-channel Downconverters
NS-3 Noise Source
IF
IF
IF
IF
IF
IF
NS-3 Noise Source
baseband
baseband
MATLAB-modelled MIMO receiver
Real-time post-ADC signal capturing (Lyrtech ADP)
IEEE 802.16e Rx

Figure 4.12: The off-line PHY-layer realization flow using GEDOMIS.

  - Load a standard channel from an existing library or build channels in MATLAB and upload them to the emulator (up to 48 taps each).

- Use the MCS RF downconverters to downconvert the RF signal; the downconversion equipment needs to be appropriately tuned to provide the IF frequency that satisfies the analogue bandwidth and sampling specifications of the ADC devices.

- Use the 8-channel VHS-ADC board to capture the post-ADC signal:

  - Since no real-time DFE is implemented it is required to apply an empirical back-off margin (e.g., 12 dB) by modifying the gain of the on-board VGAs (i.e., using a GUI) to avoid the saturation of the ADCs.
  - Capture and retrieve the data at the FPGA device of the ADC board using a preconfigured FPGA application and a dedicated GUI.

Instead of utilizing the ADP platform, it can be employed the Infiniium oscilloscope to capture data (i.e., the captures feature only 6 effective bits of resolution) or, when bypassing the channel emulation stage, to forward the RF signal to the computer running the VSA to perform standard-compliance tests.

Real-time FPGA-based PHY-layer prototyping

The GEDOMIS platform can be utilized to deploy a full real-time system and validate it using realistic mobility and fast fading channel conditions. The basic real-time PHY-layer realization flow utilizing GEDOMIS is detailed below:

- Implement the custom HDL code targeting the FPGA devices comprising the ADP:

  - Integrate both transmitter and receiver with the firmware of the boards.
  - Configure the DAC devices (e.g., filters, interpolation factor, IF signal specifications).

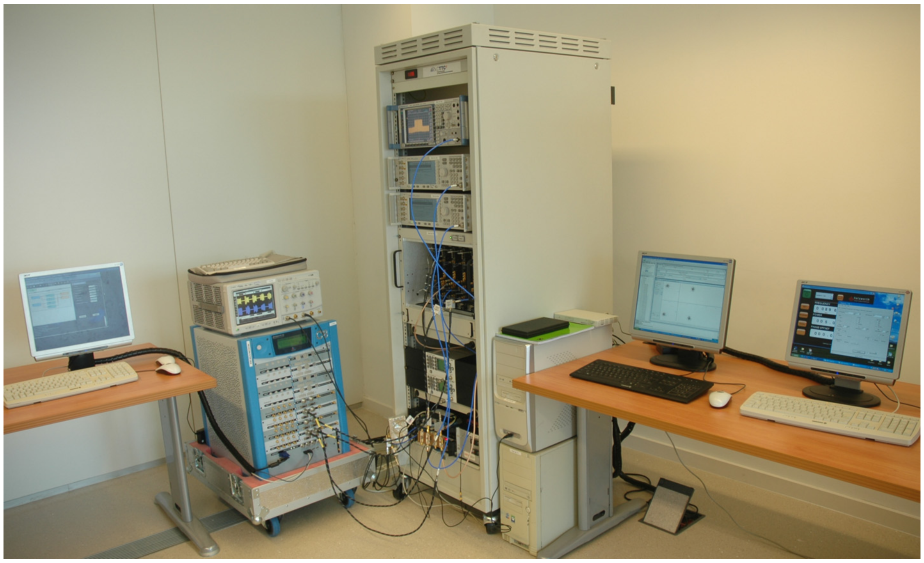


Figure 4.13: Real-time operation of the GEDOMIS testbed.

  - Implement an AGC algorithm which reconfigures on-the-fly the on-board PGAs.
  - Produce the final bitstream(s) and configure the FPGA devices.

- Use the ESG4438C VSGs to upconvert the IF signals to RF.
- Use the EB Proposim C8 channel emulator; i.e., the real-time implementation of the baseband allows the validation of both static and mobile channel models.
- The Agilent Infiniium 80000B series 4-channel oscilloscope can be utilized for data visualization, signal testing and debugging.
- The R&S spectrum analyzer can be used for RF signal testing and debugging purposes.
- The ADP can also be utilized to capture data on-the-fly (i.e., at different points of the baseband processing chain), to be latter post-processed (i.e., in MATLAB).

In Figure 4.13 it can be observed a representative setup of the GEDOMIS testbed, when it is utilized for the real-time prototyping of a broadband MIMO-OFDM system.

Chapter 5

Use Case I

Prototyping a MIMO closed-loop scheme based on the mobile WiMAX PHY-layer

5.1 Considered system

In this chapter it is detailed the design, implementation and verification of a closed-loop MIMO mobile WiMAX PHY-layer scheme, which serves as a use case of the proposed incremental development flow. The low-level details of the FPGA design are thoroughly described, to illustrate the innovative RTL solutions that were employed to efficiently handle the complex implementation of high-performance adaptive wireless communication systems.

5.1.1 Basic description of the MIMO-OFDM technology

One of the most common problems faced by designers of wireless communication systems is the phenomenon of fading that arises due the spatio-temporal variations of the wireless channel. This is inevitable in wave-reflecting and scattering environments that are subject to changes over time. The multiple received versions caused by reflections are referred to as multipath and can eventually produce a deep fade in the signal, due to a destructive combination of such received versions.

One of the main proposed techniques to tackle this effect is found in MIMO systems, which comprise multiple antennas at the transmitter and at the receiver sides. MIMO systems use diversity techniques to mitigate the effects of fading by providing multiple copies of the same signal. The use of multiple antennas dramatically reduces the probability of simultaneous deep-fades in all the receive antennas. MIMO technology may also be exploited to implement Spatial Multiplexing (SM) that significantly increases the spectral efficiency, and hence the capacity of a wireless communication system. SM is realizing high data rates by transmitting independent information streams in parallel. The latter, though, is only effective under favourable channel conditions (e.g., large SNR and rich scattering). Hence, MIMO technology features a trade-off between quality of service, provided in diversity schemes, and high data rates, provided by SM [137].

There are different ways and transmission strategies to capitalize the benefits of diversity, which primarily depend on the degree of knowledge of the channel response (i.e., CSI). In order to get such CSI at the transmitter, when channel reciprocity does not apply, a feedback channel from the receiver to the transmitter can be implemented[1]; this requires the design of a proper quantization procedure to fully exploit the limited capacity of such feedback channel. In the following, a brief survey of some MIMO transmission schemes is presented, according to the quality and quantity of the CSI available at the transmitter side[2].

An indicative MIMO transmission technique that does not require channel knowledge is the Space-Time Coding (STC), based either on block [138] (STBC) or trellis [139] (STTC) codes. It is worth to underline the Alamouti's block code [140], which uses two antennas at the transmit side and has become increasingly popular among other codes, because of its optimal and low-complexity decoding stage at the receiver; this substantial benefit of Alamouti's STBC is due to its inherent orthogonality.

When the transmitter has perfect channel knowledge, linear precoding transmission schemes are usually applied. The precoding aims at obtaining a coherent constructive combination of the multiple copies of the transmitted signal, which are received through the multiple paths of the channel [141]. Finally, the case of imperfect or incomplete CSI is also of interest, since it may correspond to the most common situations. Indeed, robust designs that take into account the presence of errors in the CSI are quite useful, since they provide schemes that are less sensitive to such errors. Some examples are [142] for single-user links, and [143] for a multi-user broadcast channel.

CSI entails key information that is used to adapt the communication link according to the instantaneous channel conditions perceived by the receiver(s) which are fed back to the transmitter. In real-life systems, such closed-loop communication schemes are made feasible when the receiver utilizes an UL transmission and the transmitter includes an UL receiver. The CSI information is used at the transmitter to modify certain PHY-layer parameters, such as the modulation scheme, the channel coding type and ratio, the allocation of subcarriers in ODFMA frames, the MIMO transmission scheme or even to apply a user-centric or system-wide energy aware policy. Adaptive wireless communication links like the one described before can yield considerable gains at the receiver, in terms of the throughput and quality of the received signal, when compared to non-adaptive systems [144–146]. Among the proposed closed-loop schemes, TAS [147, 148] has become quite popular due to its design simplicity and the resulting performance. In TAS, the knowledge of CSI at the receiver is used as an indicative metric, which is fed back at the transmitter in order

[1] In the literature it is usually assumed a reciprocal channel for those TDD systems where the coherence time of the channel is larger than the duplexing period. Nevertheless, this is not fully accurate, as in practice whilst a reciprocal channel is observed, the RF, analog and digital front-ends are not featuring an identical equivalent response for the transmission and reception chains. Accordingly, it is required either to calibrate the utilized hardware or the utilization of a dedicated feedback link.

[2] It is commonly assumed that by using training sequences or pilot tones the receiver is obtaining accurate CSI.

to change the instantaneous allocation of subcarriers per antenna (i.e., channel-aware precoding).

In wireless multipath channels it is also essential to efficiently handle the Inter-Symbol Interference (ISI) problematic. For this reason, the combination of the MIMO and OFDM technologies is widely utilized in broadband wireless communication systems. OFDM modulation converts a frequency-selective channel into a parallel collection of flat fading narrowband channels, mitigating thereby the effects of ISI.

The underlying principle of OFDM is the utilization of a guard interval which is inserted at the beginning of each OFDM symbol; this guard interval may have different lengths and essentially repeats, at the end of each symbol, part of the information located at the beginning of the symbol. This guard interval is called Cyclic Prefix (CP) and has to be long enough to accommodate the delay spread of the channel. If the mentioned condition applies, the CP converts the action of the channel on the transmitted signal from a linear convolution to a circular one. The inclusion of the CP eliminates the ISI and the transfer function is diagonalized by utilizing an IFFT at the transmitter (i.e., OFDM modulator) and an FFT at the receiver (i.e., OFDM demoulator). Nevertheless, the information carried by the CP is redundant and, hence, it results in a spectral efficiency loss. This is somehow compensated by the simplified equalization stage required at the receiver (i.e., one-tap equalizer per carrier).

Apart from the CP, there are other subcarriers included in the OFDM symbols that cannot be utilized to transmit data. For instance, a set of edge-subcarriers (denoted as guard-band subcarriers) at each side of the OFDM symbol are kept null to facilitate the digital filter design (i.e., this results in increased transition bands). Furthermore, a group of pilot tones (i.e., subcarriers with predefined values) is inserted within the OFDM symbol to facilitate the channel estimation at the receiver. Finally, the subcarrier whose frequency is equal to the RF centre frequency of the transmitter, known as DC subcarrier, is also kept null accounting for the inherent coupling with the LO that drives the operation of the analog front-end.

5.1.2 Short introduction to the Mobile WiMAX PHY-layer

The IEEE 802.16e-2005 standard [WIM, 2005], supports mobile subscriber stations at vehicular speeds and, thus, specifies a system for combined fixed and mobile BWA. The PHY layer of the mobile WiMAX supports scalable OFDMA architectures. The scalability is achieved by modifying the FFT size (i.e., 2048, 1024, 512, and 128 points), a feature that results in flexible channel bandwidths (i.e., from 1.25 to 20 MHz). Mobile WiMAX also supports AMC, various subchannelization permutation techniques, different channel coding schemes and MIMO-aided transmit-receive diversity (i.e., STBC, beamforming and SM). OFDM is used for both DL and UL transmissions.

In order to create the OFDM symbol in the frequency domain, the modulated symbols are mapped onto the subchannels that have been allocated for the transmission of the data block. According to the IEEE 802.16e-2005 standard, a subchannel is a logical collection of parallel orthogonal narrowband subcarriers. On top of it, a slot extends the subcarrier allocation to the time domain, by distributing the subchannels over sets of consecutive OFDM symbols. A slot is the minimum possible data allocation-unit defined in the IEEE 802.16e standard. Furthermore, the sets of pilot subcarriers aimed at estimating the channel at the receiver are also distributed within the defined slots.

The precise slot formation requirements depend on the mobile WiMAX permutation schemes, which basically define various subcarrier distribution modes. Each permutation is designed to serve different transmission schemes. Generally speaking, two main permutation types are defined in the standard:

- *Distributed permutations:* a pseudo-random distribution of the subcarriers is executed throughout the frequency band, which results more favourable for transmissions over frequency-selective channels when no CSI is available at the transmitter (e.g., open-loop configuration in a high-speed mobility scenario). Hence, it fits perfectly for STBC-based schemes. The two main distributed permutation schemes are the Full Usage of the Subchannels (FUSC) and the Partial Usage of the Subchannels (PUSC).

- *Contiguous permutations:* the subcarriers are distributed considering their adjacent grouping within the frequency band; this allows to exploit the available channel knowledge in order to provide the best attainable signal conditions for each user (e.g., closed-loop communication utilized in low mobility conditions). Therefore, this permutation is perfectly suiting beamforming and dynamic allocation schemes. An indicative adjacent permutation scheme for mobile WiMAX systems is the AMC.

Table 5.1 summarizes the main parameters of the permutation schemes defined in the mobile WiMAX standard. The utilized OFDMA frames may include multiple zones, hosting different permutation schemes. The only restriction is found in the Frame Control Header (FCH) and DL-MAP structures, for which the utilization of the PUSC permutation scheme is mandatory. The DL-MAP indicates the transition between zones and is composed by two OFDM symbols found at the beginning of each IEEE 802.16e frame (likewise providing the interface to the MAC layer).

PUSC permutation scheme

As indicated in Table 5.1, in PUSC a slot is generated by distributing the subchannels over two consecutive OFDM symbols. Furthermore, the subchannels are composed by 48 non-adjacent subcarriers. A series of logical structuring and permutation operations are required to construct the slots.

First, the modulated symbols are distributed among two consecutive OFDM symbols in an interleaved fashion. This means that an adjacent group of 24 complex values is allocated in the first symbol and another one is allocated in the second symbol; this process is recursively repeated until the whole capacity of both symbols is filled up. This subchannelization process continues with the next two symbols, until the whole OFDM frame is constructed (see Figure 5.1a).

In addition, each OFDM symbol is logically structured into six divisions, known as Major Groups (MGs). Then, the subcarriers are permuted independently within each MG, based on logical subsets of subcarriers named clusters (the specific implementation details of this process are given in Section 5.3.5). Finally, the clusters are also permuted according to a predefined renumbering sequence (i.e., interchanging of adjacent subcarrier groups). As a result, the sequential series of subcarriers composing the OFDM symbol before the

Scheme	Parameter (per OFDM symbol)	Value
FUSC	Data subcarriers	1536
	Pilot subcarriers	166
	Null subcarriers	346
	Subcarriers per cluster	12
	Slot size	*1 subchannel x 1 OFDM symbol*
PUSC	Data subcarriers	1440
	Pilot subcarriers	240
	Null subcarriers	368
	Major Groups (MGs)	6
	Clusters per MG	8 or 12
	Subcarriers per cluster	14
	Slot size	*1 subchannel x 2 OFDM symbols*
AMC	Data subcarriers	1536
	Pilot subcarriers	192
	Null subcarriers	320
	BINs	192
	Subcarriers per BIN	9
	Slot size	*3 BINs x 2 OFDM symbols or* *2 BINs x 3 OFDM symbols* *1 BIN x 6 OFDM symbols*

Table 5.1: Basic parameters of the IEEE 802.16e permutation schemes (for the 20 MHz bandwidth case).

permutation operations is entirely interleaved without maintaining the initial adjacency, as shown in Figure 5.2.

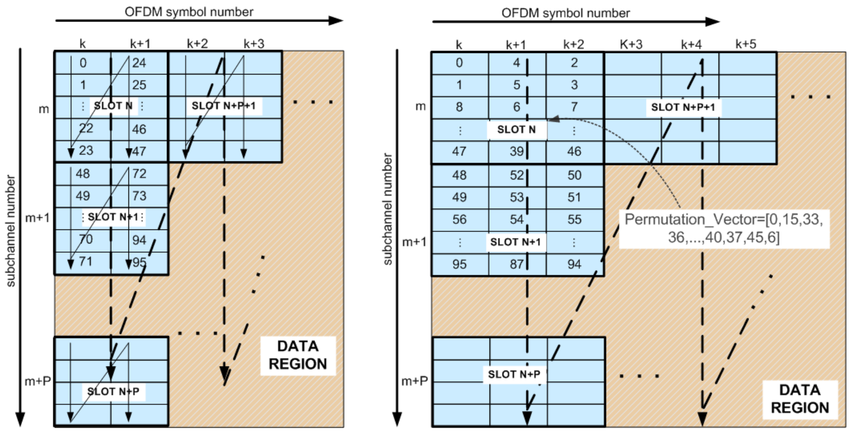
Another important aspect of the defined slots is the inclusion of the pilot subcarriers. In PUSC two pilot subcarriers are inserted to each cluster in given positions that depend on the position of the OFDM symbol within the frame (full details are provided in Section 5.2).

Each OFDM symbol is assembled by including two additional constructs. First, the (null) DC subcarrier is inserted at the central position of the OFDM symbol (e.g., 1024 in the 20 MHz bandwidth configuration), together with the guard-band subcarriers. Finally, an additional subcarrier-randomization is applied (i.e., some subcarriers are inverted), using a PseudoRandom Binary Sequence (PRBS) generator specified in the WiMAX standard. As a result, a homogeneous distribution of the transmitted power is achieved.

AMC permutation scheme

In the AMC permutation scheme, the logical structures composing the OFDM symbols are known as BINs, which include nine consecutive subcarriers. Six contiguous BINs form a subchannel. In AMC three slot configurations are defined: 3x2 (three consecutive BINs over two OFDM symbols), 2x3 (two consecutive BINs over three consecutive OFDM symbols) and 1x6 (one BIN over six OFDM symbols). The slot-formation details provided next are specific for the 2x3 configuration of the defined AMC permutation scheme.

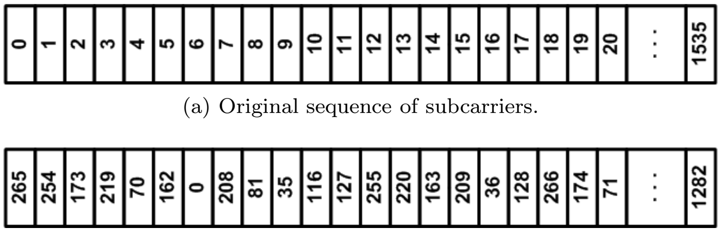
In the 2x3 AMC mode, the modulated symbols are distributed among three consecutive OFDM symbols with an order defined by a permutation vector, as shown in Figure 5.1b. More specifically for each group of 48 subcarri-



(a) PUSC scheme.

(b) 2x3 AMC scheme.

Figure 5.1: Slot creation according to the PUSC and AMC permutation schemes.



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	...	1535

(a) Original sequence of subcarriers.

265	254	173	219	70	162	0	208	81	35	116	127	255	220	163	209	36	128	266	174	71	...	1282

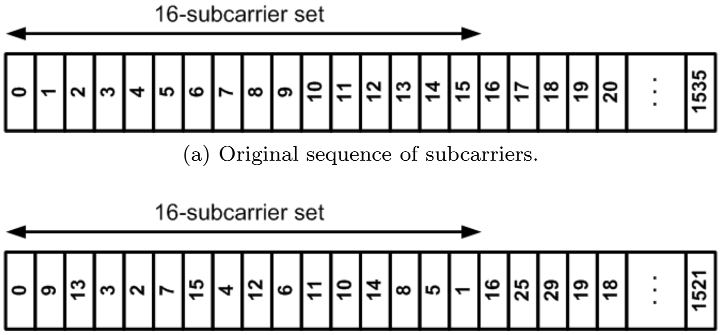
(b) Permuted subcarriers.

Figure 5.2: Subcarrier permutation in the first OFDM symbol of a PUSC slot.

ers (i.e., allocated in 2x3 BINs before the pilot insertion) the permutation vector indicates which elements are transmitted in each OFDM symbol (i.e., $quot(permutation\_vector[subcarrier\_index], 16) = OFDM\_symbol\_index$).

The resulting OFDM symbols are then permuted based on differentiated renumbering sequences, which are applied to adjacent groups of subcarriers. As a result, the sequence of subcarriers comprising the OFDM symbol before the permutation operations is partially maintained. This is due to the fact that the permutation is just modifying the internal ordering of 16-subcarrier sets (i.e., 2 BINs before the insertion of the pilot tones), which are kept adjacent for the remaining of the baseband processing blocks, as it can be observed in Figure 5.3.

Furthermore, a pilot subcarrier is inserted in each BIN at a predefined position which varies for the three OFDM symbols of the AMC slot (Section 5.2 provides full details on the specific pilot distribution schemes). Finally, exactly as in the PUSC permutation scheme, the DC and guard-band subcarriers are inserted before applying the subcarrier-randomization process.



(a) Original sequence of subcarriers.

(b) Permuted subcarriers.

Figure 5.3: Subcarrier permutation in the first OFDM symbol of a 2x3 AMC slot.

5.1.3 System specifications and included features

The scalability of the OFDM-based mobile WiMAX standard is adding a top-up complexity to the digital realization of the PHY-layer. Hence, in order to alleviate the implementation complexity and focus on the real-time implementation of the communication schemes described in Section 5.2, certain PHY-layer features related to the transmitted signal were fixed. This means that the described PHY-layer implementation includes only a subset of the flexible parameters defined in the mobile WiMAX standard. The proof-of-concept presented in this chapter considers the implementation of DL transmission and reception. The UL communication that enables the closed loop scheme is emulated. The format of the DL OFDM frames respects the WiMAX standard definition for the TDD operation. The 20 MHz channel bandwidth dramatically increased the design and implementation considerations at baseband, since it implied additional processing complexity and memory requirements. Table 5.2 summarizes the considered PHY-layer specifications.

The proof-of-concept was developed upon a point-to-point communication basis, with limited MAC functionality (emulated). The DL WiMAX signal comprises frames which encapsulate user data and silence periods that are inserted between these frames. These non-transmission periods facilitate vital processes in the receiver such as the such as the gain-adjustment (i.e., AGC) or the symbol/frame synchronization.

5.2 Utilizing an incremental development

The MIMO closed-loop communication scheme has been developed in four stages following the incremental design flow that was presented in Chapter 4. However, taking into consideration the focus of this thesis, the development of the MATLAB models of the different system configurations is not detailed per se. The principal reason supporting the previous is that the most relevant design decisions are inherently covered by describing the RTL design of the system (i.e., translation to fixed-point, consideration of hardware constraints, specifications and realistic signal conditions). Nevertheless, this section aims at describing

Parameter	Value
Wireless telecommunication standard	IEEE 802.16e-2005
Antenna schemes: SISO, SIMO, MIMO	1x1, 1x2, 2x2
Channel bandwidth (MHz)	20
Cyclic prefix (samples)	512 (1/4 of the symbol)
Modulation type	QPSK, 16/64/256-QAM
Duplex mode	TDD
FFT size	2048
OFDM symbols per frame: open-loop ||closed-loop	49 ||51
Supported permutation schemes	PUSC and 2x3 AMC (DL)
Open-loop transmission scheme	Matrix-A (Alamouti)
Closed-loop transmission scheme	TAS
ADC sampling frequency (MHz)	89.6
Baseband sampling frequency (MHz)	22.4
RF band (GHz)	2.595
IF (MHz): Transmitter ||Receiver	67.2 ||156.8
Tested channel models	ITU Ped. B (up to 3 km/h) ITU Veh. A (up to 120 km/h)

Table 5.2: Considered mobile WiMAX PHY-layer specifications and basic system parameters.

the algorithmic foundation and functionality of the baseband signal processing blocks comprising the basic RTL architecture of the developed systems (for more details regarding the MATLAB modelling utilized in the proposed methodology, please refer to [Font-Bach 12b]).

It is still relevant to detail how the developed high-level models included an abstract representation of the control plane (i.e., providing the limited MAC functionality, which for instance controlled the changes in the OFDM symbol-formatting and the generation of the synthetic user-data). Similarly, the high-level model of the system accounted for the most relevant specifications of the analog front-end of the target prototyping platform and the considered scenario (e.g., simulations utilizing experimental data or considering the quantization resulting from the DAC/ADC circuitry, the utilized RF/IF frequencies or specific channel models). The RTL design aspects related to the integration of the baseband design to the ADP boards of GEDOMIS are covered in Section 5.4; hence, Sections 5.2.1 to 5.2.4 give details of the digitally implemented signal processing logic.

5.2.1 Single-antenna open-loop scheme

The configuration parameters of the utilized DL OFDM frame are defining two data bursts with a fixed predefined format (i.e, the FCH and DL-MAP do not need to be decoded). The frame utilizes 46 OFDM symbols with user data; 28 of them are formatted according to the PUSC scheme and 18 according to the 2x3 AMC mode, as seen in Figure 5.4. Moreover, only the Quadrature Phase-Shift Keying (QPSK) modulation scheme is supported.

Single-antenna transmitter

The baseband architecture designed for the single-antenna transmitter is depicted in Figure 5.5. Following the design methodology proposed in Chapter 4,

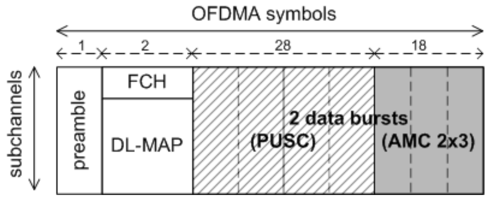


Figure 5.4: Specific frame format utilized in the single-antenna system.

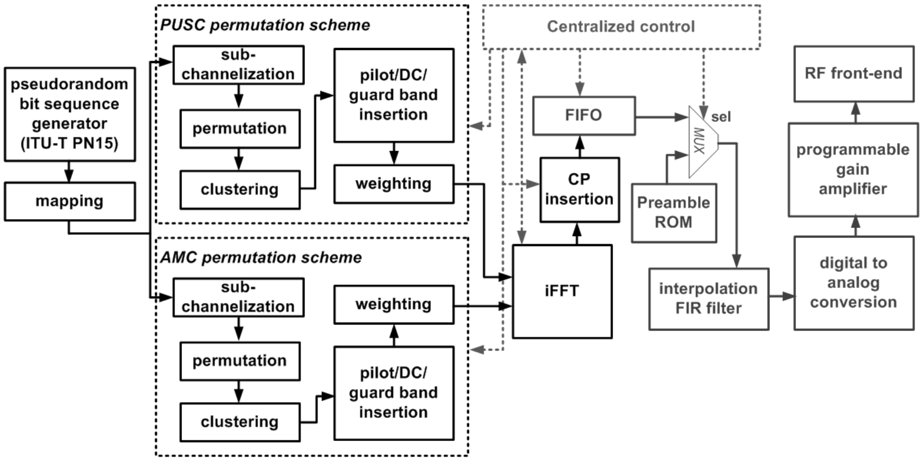


Figure 5.5: General architecture of the SISO transmitter.

a MATLAB model of the transmitter was developed first and validated using the GEDOMIS tesbed. As part of this procedure a baseline compliance of the transmitted signal was verified using the VSA software.

Bit-sequence generation and symbol mapping

Considering that the proof-of-concept targets a point-to-point system, the first processing block found in the transmitter is a PRBS generator which is based on the ITU PN15 specification [(ITU), 1996]. The PRBS generator accounts for the specified frame format and generates 135036 pseudo-random bits. Each of the 28 PUSC OFDM symbols contains 1440 subcarriers, and each of the 18 in the AMC OFDM symbols contains 1536 subcarriers. Due to the selected QPSK modulation, two bits are encoding each transmitted symbol.

Mobile WiMAX-related operations

The bulk of the processing stages in the transmitter implements the scrambling operations according to the specifications of the standard. The list of the required operations is given next:

- *Subchannelization:* the modulated symbols are distributed among the slots, according to the utilized permutation scheme. As a result, groups of related OFDM symbols are created, which contain only data subcarriers at this stage.
- *Permutation:* the I/Q values of each OFDM symbol are then scrambled

(recall that in PUSC the adjacency of the data is completely lost, while in AMC small groups remain grouped).

- *Clustering:* further scrambling is applied to groups of subcarriers (i.e., taking into account the defined logical structures within each slot).
- *Pilot, DC and guard-band insertion:* the number of required pilot and non-active subcarriers are inserted in the indicated positions within each logical structure (i.e., cluster and BIN).
- *Weighting:* the standard defines a PRBS generator which is used to change the sign of a subset of the I/Q values included in each OFDM symbol.

To implement the mentioned operations, which are devoted to reordering the data at each processing stage, a resource-efficient and minimized-latency memory structure has been designed, as detailed in Section 5.3.4.

IFFT and CP insertion

The IFFT converts the generated OFDM symbols from the frequency domain to the time domain. A CP is then inserted at the beginning of each OFDM symbol, repeating its last 512 samples. The CP helps to mitigate the effects of ISI and facilitates the time synchronization at the receiver.

Insertion of the preamble

After the insertion of the CP at each OFDM symbol containing user data, it is still required to add a preamble (i.e., an additional OFDM symbol must be inserted to facilitate the synchronization process at the receiver).

Single-antenna receiver

An indicative representation of the baseband processing blocks encountered at the receiver is shown in Figure 5.6. Before entering in details it is required first to define the model of the received signal, which has to account for the impairments featured in the target prototyping platform (described in Section 4.4.5). Hence, the received signal before the ADC stage can be expressed as:

$$c(t) = \Re\{x(t) \cdot e^{j2\pi(f_{IF}+\Delta f)t}\} + A + B \cdot cos(2\pi(f_{IF} + \Delta f)t + \varphi) + w(t), \quad (5.1)$$

where $x(t)$ represents the useful part of the received baseband signal, f_{IF} is the IF, Δf is the CFO, A is the DC level introduced by the baseband boards' chassis, $B \cdot cos(2\pi(f_{IF} + \Delta f)t + \varphi)$ represents the unwanted residual carrier, located at the center of the useful signal-spectrum (i.e., introduced by LO coupling at the transmitter) and, finally, $w(t)$ is the zero-mean white circularly symmetric Gaussian noise. The useful part of the received baseband signal can be expressed as follows:

$$x(t) = \tilde{x}(t) * H(t), \quad (5.2)$$

where $\tilde{x}(t)$ is the equivalent transmitted baseband signal and $H(t)$ is the equivalent baseband representation of the time impulse response of the SISO channel between the transmit and receive antennas, in respect to the center RF frequency, $f_{RF} + \Delta f$.

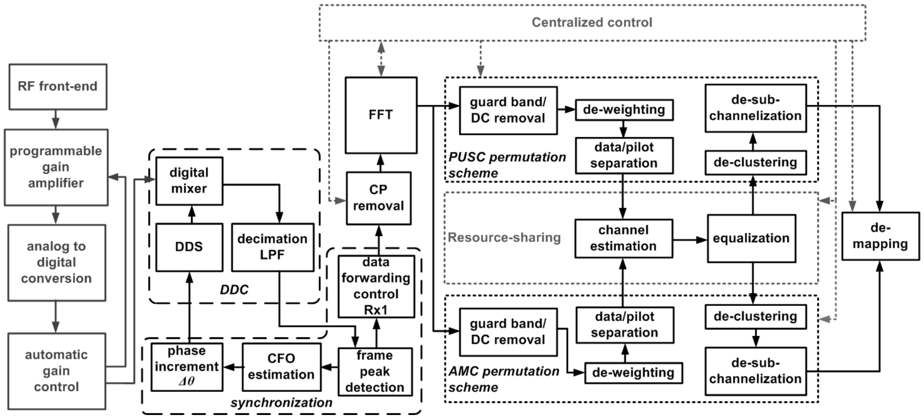


Figure 5.6: Block-diagram of the implemented SISO receiver.

AGC and DDC

The first digital processing block of any wireless communication receiver is the AGC, which timely adapts the power-level of the post-ADC signal to take full advantage of the system's dynamic range. Its specific design and implementation details depend on the format of the transmitted frame and the specific signal-acquisition hardware being utilized, since the AGC implements an algorithm that controls dedicated gain-adjustment circuitry (i.e., PGA).

Similarly, the specific requirements of the translation of the received signal to baseband are also related to the specifications of the target analog front-end. For instance, in 0-IF downconverter ICs the sampled signal is already in baseband. In the presented use case, given the characteristics of the GEDOMIS testbed (i.e., RF-to-IF downconverters are employed), a DDC is effectively required after the AGC processing block to provide the complex baseband representation of the digitalized signal. More specifically, the DDC is in charge of the channel frequency translation, the I/Q component extraction and the CFO correction. The DDC utilizes a DDS and a digital-mixer to translate any frequency band within the analog bandwidth of the ADCs down to zero frequency (i.e., baseband). Ideally, the output frequency of the DDS, f_{DDS}, should be tuned at 22.4 MHz, which is the baseband sampling frequency defined in the WiMAX standard. In practice though, this value will be altered in the presence of CFO, as indicated below:

$$f_{DDS} = 22.4 + \Delta f \ MHz, \tag{5.3}$$

Δf represents the CFO that is usually defined in terms of the separation between adjacent subcarriers of the OFDM modulation:

$$\Delta f = \alpha \frac{22.4 \cdot 10^6}{2048}, \tag{5.4}$$

where α is the CFO normalized with respect to the intercarrier separation (i.e., in practice the CFO will not be higher than one half the intercarrier separation, $\alpha \in (-0.5, 0.5)$), 22.4 MHz is the baseband sampling frequency and 2048 is the FFT size (i.e., corresponding to a 20 MHz bandwidth).

The full details regarding the operation of the analog front-end and the implemented AGC and DDC stages are introduced in Section 5.4.

Synchronization, CFO estimation and correction

The DL frame comprises several OFDM symbols. In the open-loop mode 46 of them are used for the user-data; each OFDM symbol is having 2560 samples (i.e., including a 512-sample CP). During the inter-frame silence periods the receiver is continuously monitoring the incoming signal in order to detect the beginning of the following frame using a synchronization algorithm. In more detail, the symbol detection is required to properly locate the FFT-window of the samples comprising each OFDM symbol. In a TDD system like the one defined in the IEEE 802.16e standard, this is feasible with the inclusion of a CP at the beginning of each OFDM symbol (i.e., it results in a cyclical repletion in the received signal that can be detected by performing a cross-correlation).

Taking into account the delay spread of the selected channel models (e.g., ITU Vehicular A at 60 km/h), a few samples in the CP need to be discarded to avoid an unreliable operation of the FFT window-locator. After several simulations utilizing realistic channel data, it has been decided that only 455 out of the 512 samples in the CP can be used for the timing synchronization. Hence, the implemented synchronization technique is based on a sliding window of 2048+455 samples, which allows us to calculate the cross-correlation of two groups of 455 samples (having a separation of 2048 samples). The expression corresponding to the square of the correlation when the sliding window starts at the nth sample is given by:

$$|r_s[n]|^2 = \frac{|\sum_{l=0}^{454} s^*[n+l] \cdot s[n+l+2048]|^2}{(\sum_{l=0}^{454} |s[n+l]|^2) \cdot (\sum_{l=0}^{454} |s[n+l+2048]|^2)}, \tag{5.5}$$

where $s[n]$ is the equivalent complex baseband signal at the output of the DDC, sampled at 22.4 MHz.

A peak in $|r_s[n]|^2$, indicates the detection of the symbol and thus the sample where the CP starts, i.e., $pos_{cp} = \arg\ \max_n |r_s[n]|^2$. Additionally the phase of the correlation - or equivalently, the phase of the numerator of $r_s[n]$, which is given by $\sum_{l=0}^{454} s^*[n+l] \cdot s[n+l+2048]$ - can be used to estimate the phase shift of the received signal in the presence of CFO. Using the notation given in (5.4), the phase shift between two signal samples delayed by 2048 positions is equal to $e^{j2\pi\Delta ft}|_{t=2048\frac{1}{22.4\cdot 10^6}} = e^{j2\pi\alpha}$. Therefore, the estimated CFO or, equivalently, α can be defined as:

$$\alpha = \frac{1}{2\pi}\measuredangle(r_s[pos_{cp}]), \tag{5.6}$$

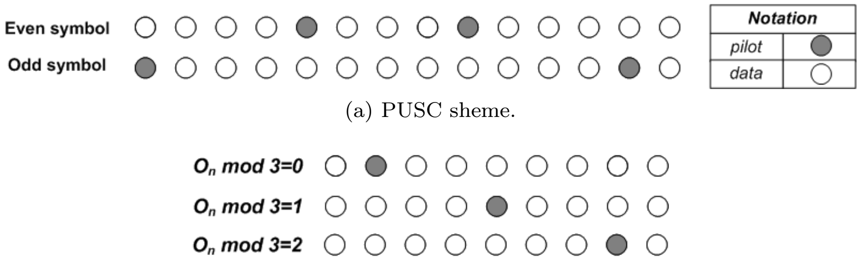
where α can be calculated using a CORDIC algorithm.

CP removal and FFT

Before the OFDM symbols are further processed it is required first to remove the CP. Hence, the first 512 samples of each set of 2560 are removed. Then an FFT is applied to each resulting group of 2048 samples, transforming the received signal from the time to the frequency domain. According to the utilized notation, this can be expressed as $S[k] = FFT(s[n])$.

Guard-band and DC removal

The DC and guard-band subcarriers are removed at the output of the FFT. The resulting subset, $S[k]$, is thus solely composed by active subcarriers (i.e.,



(a) PUSC sheme.

(b) 2x3 AMC scheme.

Figure 5.7: Pilot distributions utilized in the single-antenna system.

$k \in \{0, ..., n_U - 1\}$, where n_U represents the number of subcarriers used to transmit user-data and pilot tones). In the PUSC structured OFDM symbols, out of the 2048 subcarriers available, 1440 are used for data transmission and 240 for pilot tones transmission ($n_U = 1679$). Similarly 1536 data and 192 pilot subcarriers are used in AMC ($n_U = 1727$). Additionally, the channel estimation processing block that follows requires the subcarrier randomization that was applied at the transmitter to be reversed (i.e., de-weighting process).

Pilot extraction and channel estimation

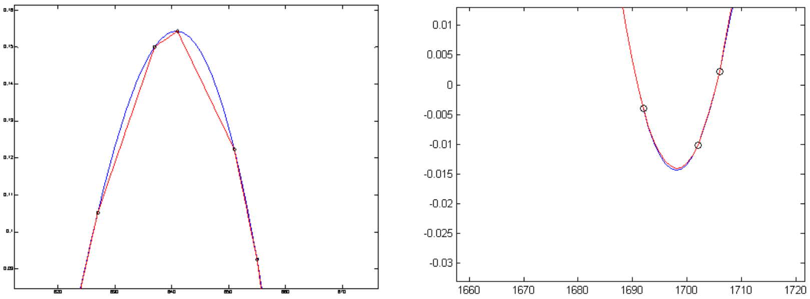
The channel estimation in the receiver is based on the pilot subcarriers that are being transmitted in each OFDM symbol. The IEEE 802.16e standard defines the predefined I/Q values (i.e., $\frac{4}{3} + 0i$ is the single value used in our system configuration) and location (i.e., frequency), p_k, of such special subcarriers (i.e., $p_k \in \{0, ..., n_U - 1\}$). The number and distribution of pilot tones depends on the subcarrier permutation scheme.

As already mentioned, when the PUSC permutation scheme is used, clusters of 14 contiguous subcarriers are defined, with two of them used transmitting pilot tones. The specific cluster structure utilized for the single-antenna configuration is shown in Figure 5.7a. As it can be observed, the pilot positions depend on the parity of the OFDM symbol index. Similarly, in the AMC scheme the pilot distribution is defined by the BIN structures. More specifically, one pilot subcarrier is introduced at each BIN. Three possible locations are defined, as shown in Figure 5.7b, where $O_n \in \{0, ..., 47\}$ is the index of the transmitted OFDM symbol (i.e., location within the frame). Finally, the described structures are repeated for each formatted slot in the OFDM frame.

To estimate the channel between the transmit and receive antennas, first the channel frequency response at the pilot tones, $\tilde{H}[p_k]$, is calculated as follows:

$$\tilde{H}[p_k] = \frac{S[p_k]}{\frac{4}{3}}, \tag{5.7}$$

where $S[p_k]$ represents the output of the FFT at the position coresponding to the kth pilot tone. Thus, $\tilde{H}[p_k]$ is a discrete function calculating the channel frequency response at the pilot tones. An interpolation of the pilot positions is then required to estimate the channel at the frequencies where data subcarriers were transmitted. After comparing different algorithms in MATLAB (see Figure 5.8), a second order Newton interpolating polynomial was selected, which provides an acceptable trade-off between accuracy and implementation complexity



(a) Linear interpolation: accounts for two neighbour pilots, resulting in a notble estimation error.

(b) Quadratic interpolation: uses three neighbouring pilots, providing a precise estimation.

Figure 5.8: Early algorithm selection for the channel estimation stage.

(accounting for the number of pilot tones in each OFDM symbol and the considered channel specifications). Thus, the channel frequency response for the data subcarrier corresponding to the kth position is calculated as follows:

$$\begin{aligned}\tilde{H}[k] =&\tilde{H}[p_{c_1}] + \frac{\tilde{H}[p_{c_2}] - \tilde{H}[p_{c_1}]}{p_{c_2} - p_{c_1}} \cdot (k - p_{c_1}) \\ &+ \frac{\frac{\tilde{H}[p_{c_3}]-\tilde{H}[p_{c_2}]}{p_{c_3}-p_{c_2}]} - \frac{\tilde{H}[p_{c_2}]-\tilde{H}[p_{c_1}]}{p_{c_2}-p_{c_1}}}{p_{c_3} - p_{c_1}} \cdot (k - p_{c_1}) \cdot (k - p_{c_2}),\end{aligned} \tag{5.8}$$

where p_{c_r} represents the location of each of the three closest neighbouring pilot tones to $S[k]$, for every $r \in \{1, 2, 3\}$.

Equalization

The originally transmitted symbols can be estimated at this stage utilizing the estimated channel as follows:

$$\hat{d}_k = \frac{S[k]}{\tilde{H}[k]}, \tag{5.9}$$

De-clustering and de-subchannelization

At this stage, it is required to recover the original sequence of transmitted symbols, which means that the standard-related operations are basically implemented here. A de-scrambling of the basic logic structures is first required (i.e., de-clustering). Then, the de-subchannelization stage decomposes the reordered slots taking into account the intricacies of both the PUSC and AMC permutation schemes, as fully described in Section 5.3.4.

De-mapper

Since all symbols are modulated using the QPSK scheme, it is sufficient to inspect the sign of the real and imaginary components to recover the originally transmitted bit sequence.

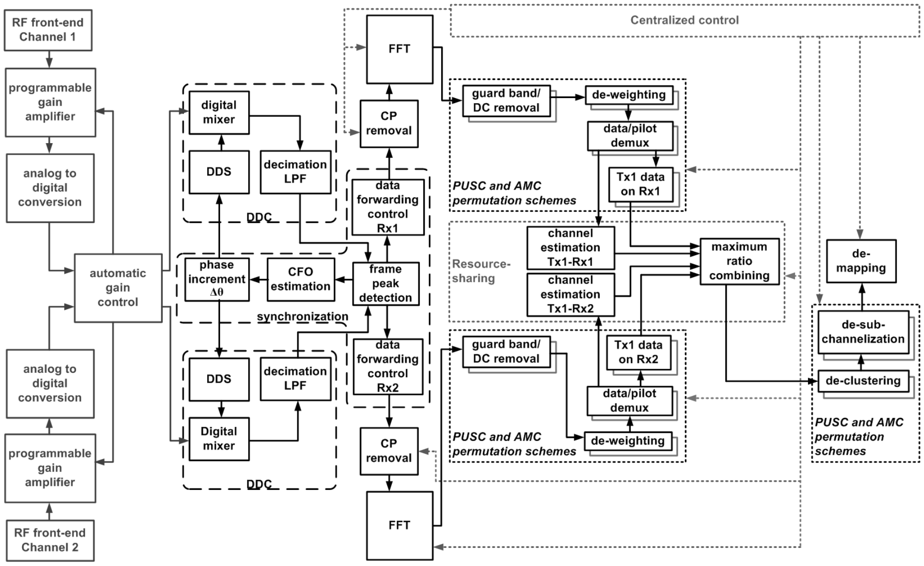


Figure 5.9: General architecture of the 1x2 SIMO receiver.

5.2.2 1x2 Single Input Multiple Output (SIMO) open-loop scheme

The second milestone in the incremental development flow was a 1x2 SIMO PHY-layer scheme. The signal generation made use of the single-antenna transmiter described before, whereas the design of the receiver required to be extended, as shown in Figure 5.9. The processing stages from the CP removal to the channel estimation are simply replicated for each receive-antenna processing branch. The same applies to the DSP logic blocks between the de-clustering and de-mapping processes. Hence, the required extension is basically found on the synchronization and symbol decoding processing elements, as it is described in the following.

The received signal before the ADC stage at the ith receive antenna can be expressed as:

$$c_i(t) = \Re\{x_i(t) \cdot e^{j2\pi(f_{IF}+\Delta f)t}\} + A_i + B_i \cdot cos(2\pi(f_{IF}+\Delta f)t + \varphi_i) + w_i(t), \quad (5.10)$$

where $x_i(t)$ represents the useful part of the received baseband signal at the ith receive antenna, which is defined as follows:

$$x_i(t) = \tilde{x}(t) * H_i(t), \quad (5.11)$$

where $H_i(t)$ is the equivalent baseband representation of the time impulse response of the SIMO channel between the single transmit antenna and the ith receive antenna, in respect to $f_{RF} + \Delta f$.

Synchronization, CFO estimation and correction

The correlation of the synchronization algorithm (5.5) was adapted to the two-antenna configuration as follows:

$$|r_s[n]|^2 = \frac{|\sum_{i=1}^{n_R} \sum_{l=0}^{454} s_i^*[n+l] \cdot s_i[n+l+2048]|^2}{(\sum_{i=1}^{n_R} \sum_{l=0}^{454} |s_i[n+l]|^2) \cdot (\sum_{i=1}^{n_R} \sum_{l=0}^{454} |s_i[n+l+2048]|^2)}, \quad (5.12)$$

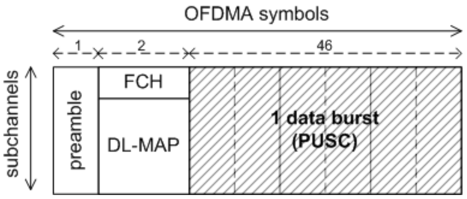


Figure 5.10: Frame format utilized in the open-loop MIMO system.

where $s_i[n]$ is the equivalent complex baseband signal at the output of the DDC, sampled at 22.4 MHz, at the ith receive antenna processing branch, and n_R denotes the number of receive antennas (i.e., $n_R = 2$ in our case). The full details of the designed architecture are provided in Section 5.3.1.

Maximum Ratio Combining

In the SIMO configuration, diversity combining is applied in the receiver in order to maximize the overall SNR and, thus, improve the radio link performance. The receive diversity is exploited by using MRC. Up to the channel estimation, the received symbols are processed by independent baseband processing branches. The resulting outputs at each branch (two in our case) are combined by weighting them according to the signal amplitude, as indicated hereafter:

$$\hat{d}_k = \frac{\sum_{i=1}^{n_R} S_i[k] \cdot \tilde{H}_i^*[k]}{\sum_{i=1}^{n_R} |\tilde{H}_i[k]|^2} \tag{5.13}$$

where $\hat{d}_k$ represents the estimate of the symbol transmitted through the kth carrier (i.e., kth input of the IFFT at the transmitter), $S_i[k]$ represents the output of the FFT corresponding to the kth carrier at the ith receive antenna, and $\tilde{H}_i[k]$ is the estimate of the channel frequency response for such carrier between the single transmit antenna and the ith receive antenna. Further details regarding the efficient implementation of the MRC technique (when combined with a Space-Time Decoding, STD, scheme) are provided in Section 5.3.3.

5.2.3 2x2 MIMO STBC-based system

The first difference when having two transmit antennas is found in the frame format, which features a single PUSC-formatted data burst (i.e., comprising 46 OFDM symbols), as indicated in Figure 5.10. The selected encoding scheme is Alamouti's STBC (defined as matrix A in the WiMAX standard). The preamble is only found in the first transmit antenna (i.e., the second transmit antenna is in silence when the first antenna transmits the preamble).

The 2x2 MIMO scheme requires to extend the functionality of various processing blocks in the transmitter and the receiver as described in the following sections.

Multi-antenna transmitter

Figure 5.11 shows the signal processing architecture designed for the multi-antenna transmitter. Most of the processing blocks are basically a scaled version of those designed for its single-antenna counterpart, considering for instance

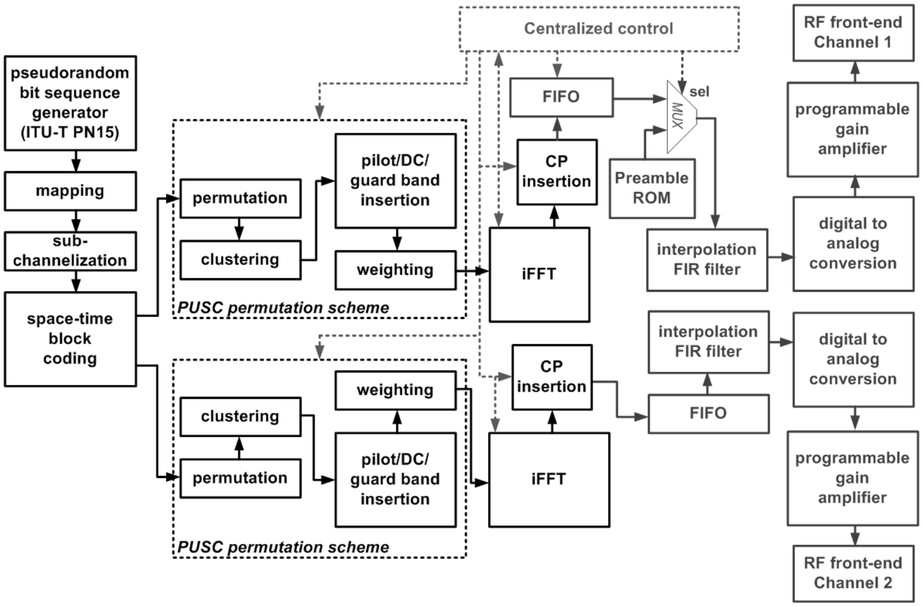


Figure 5.11: Baseband architecture of the MIMO transmitter.

that two data flows need to be processed. The main difference resides in the STBC block, which is detailed below.

STBC

As briefly introduced before, the open-loop MIMO transmitter is based on an Alamouti's STBC scheme. Hence, the subcarrier allocation for each transmit antenna is dictated by the matrix A configuration specified in the mobile WiMAX standard. More specifically, for the case of two transmit antennas the allocation matrix is defined as:

$$A = \begin{bmatrix} d_N & -d^*_{N+1} \\ d_{N+1} & d^*_N \end{bmatrix}, \tag{5.14}$$

where $d_1, d_2, ..., d_{66240}$ is the single stream of complex symbols at the output of the subchannelization block (i.e., accounting for the specified frame format); the columns represent consecutive OFDM symbols (i.e., time) and the rows represent the transmit antennas (i.e., spatial streams). The design and implementation of the Alamouti STBC scheme is discussed in Section 5.3.4.

Multi-antenna receiver

The different baseband processing stages of the MIMO receiver are shown in Figure 5.12. Considering that the underlying SISO system had already been designed, the MIMO system was extended or modified incrementally, especially in respect to the channel estimation and Space Time Block Decoding (STBD) processing blocks. Moreover it is worth mentioning that design of the DFE is identical with one conducted for the SIMO system.

The received signal before the ADC stage at the ith receive antenna can be described with the following equation:

$$c_i(t) = \Re\{x_i(t) \cdot e^{j2\pi(f_{IF}+\Delta f)t}\} + A_i + B_i \cdot cos(2\pi(f_{IF}+\Delta f)t + \varphi_i) + w_i(t), \tag{5.15}$$

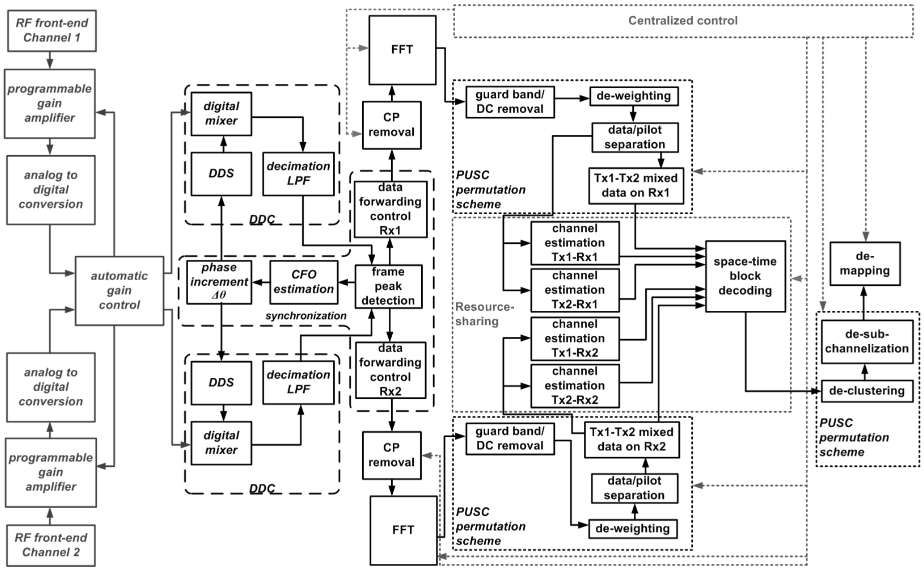


Figure 5.12: Baseband architecture of the 2x2 MIMO STBC-based receiver.

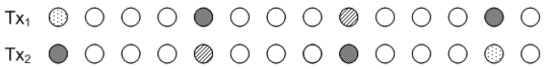
where $x_i(t)$ is the useful part of the received baseband signal at the ith receive antenna that can be modelled as:

$$x_i(t) = \sum_{j=1}^{n_T} \tilde{x}_j(t) * H_{i,j}(t), \tag{5.16}$$

where $\tilde{x}_j(t)$ is the equivalent baseband signal transmitted from the jth transmit antenna, with n_T being the number of assumed transmit antennas (in this case $n_T = 2$), and $H_{i,j}(t)$ is the equivalent baseband representation of the time impulse response of the MIMO channel between the jth transmit antenna and the ith receive antenna, in respect to $f_{RF} + \Delta f$.

Pilot extraction and channel estimation

As in the single-antenna case, the specific pilot distribution depends on the utilized permutation scheme. In the multi-antenna scheme, the locations of the pilot carriers are defined for each transmit antenna; for the jth transmit antenna, it is defined $p_{k,j} \in \{0, ..., n_U - 1\}$. Considering that only the PUSC scheme is featured in this MIMO system, two pilots are included at each cluster (their values are the same utilized in the SISO system). Figure 5.13 shows the detailed cluster structure for the 2x2 MIMO configuration (i.e., $n_T = 2$). As it can be observed, the PUSC permutation scheme distributes the pilot tones for each antenna in two consecutive OFDM symbols. This implies that the channel estimation is applied in pairs of consecutive OFDM symbols (i.e., the estimated channel frequency response is assumed to be the same for both). Taking into account the configuration of the slots, this cluster structure is then cyclically repeated each four OFDM symbols, with $O_n \in \{0, 1, ..., 23\}$ being the index of the transmitted OFDM symbol pair within the frame. Therefore, the cluster structure defines $p_{k,j}$ for each OFDM symbol and each transmit antenna j. When a subcarrier k is used in a given OFDM symbol to transmit a



subcarrier function	notation	O_n mod 4
data / pilot / null	●	2,3 / 1 / 0
	▨	2,3 / 0 / 1
	◌	0,1 / 3 / 2
	●	0,1 / 2 / 3
data	○	0,1,2,3

Figure 5.13: Distribution of the pilot tones in the MIMO PUSC configuration.

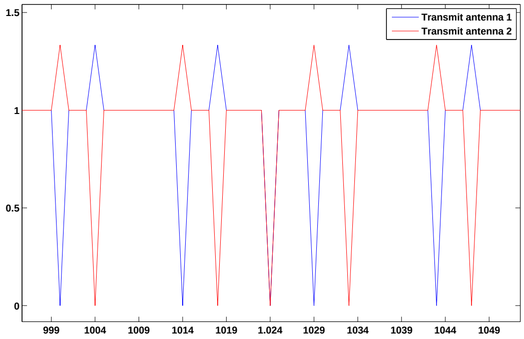


Figure 5.14: Spectrum of the transmitted STBC signal, where it can be observed the DC and various pilot subcarriers.

pilot through one of the two transmit antennas, the equivalent subcarrier in the OFDM symbol of the other antenna remains in silence (i.e., null subcarrier), and vice-versa, as shown in Figure 5.14. This avoids having interference in the pilot positions.

Each processing branch of the MIMO-enabled receiver has to estimate the corresponding channels from all the transmit/receive antenna-pairs. To estimate the channel frequency response for the pilot subcarriers, $\tilde{H}_{i,j}[p_{k,j}]$, the following formulation is applied:

$$\tilde{H}_{i,j}[p_{k,j}] = \frac{S_i[p_{k,j}]}{\frac{4}{3}}, \tag{5.17}$$

where $S_i[p_{k,j}]$ represents the output of the FFT at the jth transmit antenna corresponding to the position of the kth pilot tone transmitted from the ith transmit antenna, with $j \in \{1, 2\}$ in our case, and $\tilde{H}_{i,j}[p_{k,j}]$ being the discrete representation of the channel frequency response at the pilot tones between the ith receive antenna and the jth transmit antenna. Then, the second order polynomial interpolation (5.8) is updated accordingly:

$$\begin{aligned}\tilde{H}_{i,j}[k] =& \tilde{H}_{i,j}[p_{c_1,j}] + \frac{\tilde{H}_{i,j}[p_{c_2,j}] - \tilde{H}_{i,j}[p_{c_1,j}]}{p_{c_2,j} - p_{c_1,j}} \cdot (k - p_{c_1,j}) \\ &+ \frac{\frac{\tilde{H}_{i,j}[p_{c_3,j}] - \tilde{H}_{i,j}[p_{c_2,j}]}{p_{c_3,j} - p_{c_2,j}} - \frac{\tilde{H}_{i,j}[p_{c_2,j}] - \tilde{H}_{i,j}[p_{c_1,j}]}{p_{c_2,j} - p_{c_1,j}}}{p_{c_3,j} - p_{c_1,j}} \cdot (k - p_{c_1,j}) \cdot (k - p_{c_2,j}),\end{aligned} \tag{5.18}$$

where $p_{c_r,j}$ represents the location of one of the three neighbouring pilot tones,

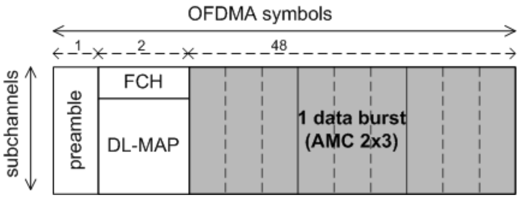


Figure 5.15: Format of the OFDM frame utilized in the closed-loop MIMO system.

of the transmit antenna j, closest to the kth carrier, for each $r \in \{1, ..., 3\}$. The complete design details of the channel estimation for the 2x2 MIMO open-loop system are provided in Section 5.3.2.

Matrix A STBD

The applied STBC transmission scheme encodes data symbols, d_k, in pairs and distributes them in groups of two OFDM symbols (denoted as matrix A). In order to decode and estimate the transmitted data symbols, the following operations are applied:

$$\hat{d}_k[2O_n] = \frac{\sum_{i=1}^{n_R} \tilde{H}_{i,1}^*[k] \cdot S_i[k, 2O_n] + \tilde{H}_{i,2}[k] \cdot S_i^*[k, 2O_n + 1]}{\sum_{i=1}^{n_R} |\tilde{H}_{i,1}[k]|^2 + |\tilde{H}_{i,2}[k]|^2}, \tag{5.19}$$

$$\hat{d}_{k+1}[2O_{n+1}] = \frac{\sum_{i=1}^{n_R} \tilde{H}_{i,2}^*[k] \cdot S_i[k, 2O_n] - \tilde{H}_{i,1}[k] \cdot S_i^*[k, 2O_n + 1]}{\sum_{i=1}^{n_R} |\tilde{H}_{i,1}[k]|^2 + |\tilde{H}_{i,2}[k]|^2}, \tag{5.20}$$

where O_n and $O_n + 1$ represent the indexes of the two consecutive OFDM symbols that are jointly encoded. Note that in (5.19) and (5.20) it is assumed that the gain applied by the AGC to the incoming sample-streams is equal for both receive antennas. The joint design and implementation of the STBD and MRC operations is discussed in Section 5.3.3.

5.2.4 Extending the design to a 2x2 MIMO closed-loop scheme

The closed-loop MIMO scheme is the last incremental development stage presented in this chapter. The frame format was adapted to reintroduce the AMC permutation scheme, as shown in Figure 5.15, considering that the featured adjacent distribution of subcarriers fits perfectly the requirements of a closed-loop technique. Moreover, the PUSC-formatted OFDM symbols (i.e., DL-MAP) are just utilized to describe the precise allocation of subcarriers of each frame. Given the specific slot configuration employed in the 2x3 AMC permutation scheme, the number of symbols utilized to transmit user data needs to be extended to 48. This results in a frame comprising 51 OFDM symbols (including the preamble). It should be clarified that, while the frame format utilizes a fixed number of OFDM symbols for each communication scheme (i.e., 2 for the open-loop and 48 for the closed-loop), the designed baseband architecture is ready to accept frames using any other combination of data-bursts.

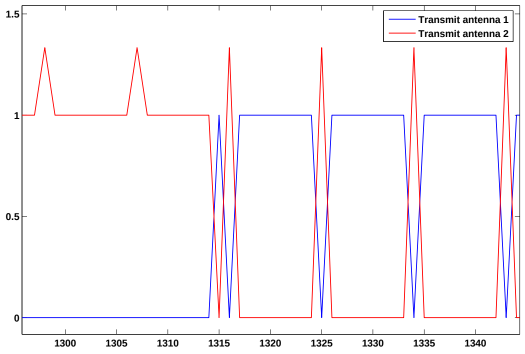


Figure 5.16: Spectrum of the transmitted TAS signal.

In the proposed closed-loop configuration, the subcarrier allocation utilized in the AMC-structured data burst depends on the CSI provided by the receiver to the transmitter. The feedback information is procured in a per received-frame basis (i.e., it is defining the subcarrier allocation for the next frame). Consequently, this information is strictly related to the formation-process of the DL-MAP. As a result, the PHY-layer provided both open-loop (i.e., PUSC zone) and closed-loop (i.e., AMC-zone) functionalities.

A TAS scheme was selected for the closed-loop configuration. According to the AMC permutation scheme, each of the 48 OFDM symbols in the data burst is comprising 96 groups of 16 adjacent subcarriers. Taking advantage of this, a transmit antenna is selected for each 16-subcarrier group (i.e., the equivalent subcarriers in the OFDM symbol of the other antenna remain in silence, as shown in Figure 5.16). Furthermore, for the sake of an improved receiver performance, the transmission of pilot subcarriers is not affected by the antenna selection mechanism; hence, the utilized pilot distribution remains as described in the standard for the 2x2 MIMO open-loop AMC permutation scheme (indicatively, in the figure all pilots are found in the second transmit antenna regardless of the selected transmit antenna).

The basic operation of the proposed TAS scheme is shown in Figure 5.17. From the transmitter point of view, the received feedback defines the dynamic subcarrier allocation process, as well as the contents of the PUSC symbols composing the DL-MAP information. On the receiver side, these two OFDM symbols need to be decoded in order to correctly process the dynamically adapted AMC data burst (i.e., precise information regarding the subcarrier allocation utilized in each AMC-formatted OFDM symbol is required). Moreover, the receiver also needs to provide meaningful information to the transmitter regarding the quality of the channel perceived in each of the two receive antennas for every 16-subcarrier group. The latter is based on the calculation of an appropriate signal-quality metric.

Accounting for the point-to-point communication link realized in the prototype, the metric calculated at the receiver was directly utilized to select a transmit antenna for each subcarrier group. In other words, the TAS-decision was performed at the receiver and communicated, via the dedicated feedback

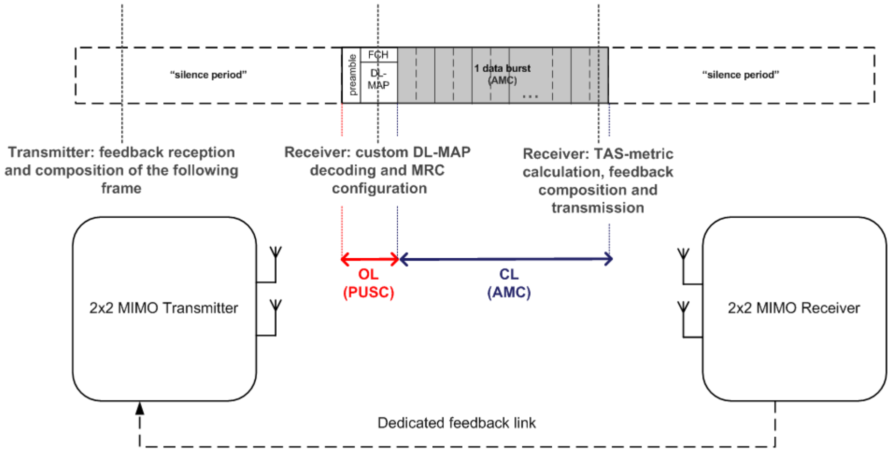

"silence period"
preamble
FCH
DL-MAP
1 data burst (AMC)
...
"silence period"
Transmitter: feedback reception and composition of the following frame
Receiver: custom DL-MAP decoding and MRC configuration
Receiver: TAS-metric calculation, feedback composition and transmission
2x2 MIMO Transmitter
OL (PUSC)
CL (AMC)
2x2 MIMO Receiver
Dedicated feedback link

Figure 5.17: Time-domain diagram of the basic operation of the TAS scheme.

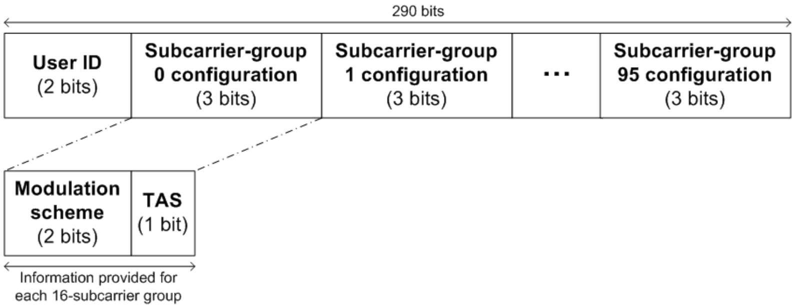

290 bits
User ID (2 bits)
Subcarrier-group 0 configuration (3 bits)
Subcarrier-group 1 configuration (3 bits)
...
Subcarrier-group 95 configuration (3 bits)
Modulation scheme (2 bits)
TAS (1 bit)
Information provided for each 16-subcarrier group

Figure 5.18: Designed feedback-information structure.

channel, to the transmitter (which was only adapting the signal according to the received feedback information).

The designed feedback structure, shown in Figure 5.18, required only 290 bits and enabled the inclusion of additional PHY-layer functionalities. Indicatively, the modulation could be defined in a per-frame basis, by selecting to use QPSK, 16-, 64- or 256-Quadrature Amplitude Modulation (QAM). Furthermore, only one user was considered in the developed system (i.e., a fixed user ID value was utilized).

Multi-antenna closed-loop transmitter

The different baseband processing blocks of the MIMO closed-loop transmitter are shown in Figure 5.19. Besides the evident extension of the DSP architecture described for its MIMO open-loop counterpart to reintroduce the AMC permutation scheme, the core functional-evolution of the designed baseband is found in the provision of run-time adaptivity.

Bit-sequence generation and programmable symbol mapping

The previously introduced PRBS generator was extended, accounting for the variable symbol-modulation scheme (i.e., the precise length of the bit sequence to be generated depends on the received feedback information). In more details,

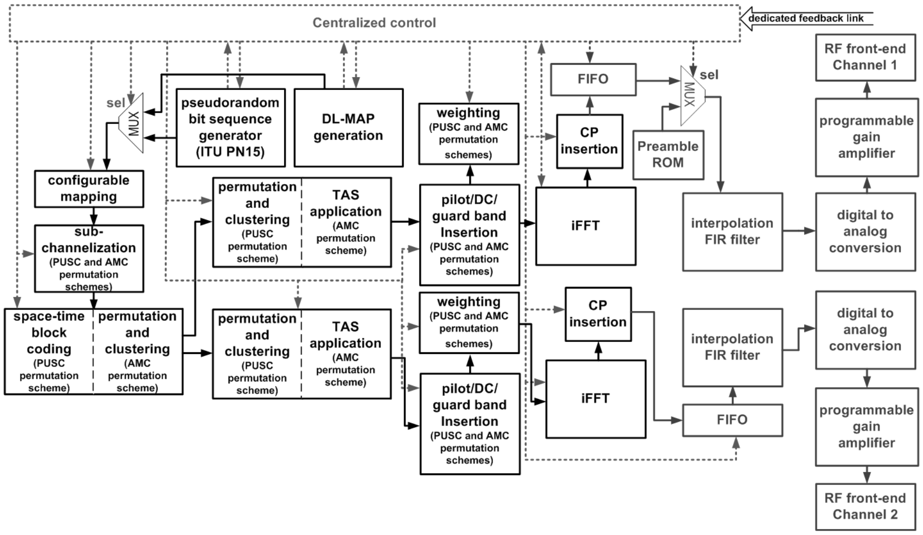

Centralized control
dedicated feedback link
RF front-end Channel 1
FIFO
sel
MUX
sel
MUX
pseudorandom bit sequence generator (ITU PN15)
DL-MAP generation
weighting (PUSC and AMC permutation schemes)
CP insertion
Preamble ROM
programmable gain amplifier
configurable mapping
permutation and clustering (PUSC permutation scheme)
TAS application (AMC permutation scheme)
pilot/DC/ guard band Insertion (PUSC and AMC permutation schemes)
iFFT
interpolation FIR filter
digital to analog conversion
sub-channelization (PUSC and AMC permutation schemes)
space-time block coding (PUSC permutation scheme)
permutation and clustering (AMC permutation scheme)
permutation and clustering (PUSC permutation scheme)
TAS application (AMC permutation scheme)
weighting (PUSC and AMC permutation schemes)
CP insertion
interpolation FIR filter
digital to analog conversion
pilot/DC/ guard band Insertion (PUSC and AMC permutation schemes)
iFFT
FIFO
programmable gain amplifier
RF front-end Channel 2

Figure 5.19: General architecture of the MIMO closed-loop transmitter.

each modulated symbol required the generation of N bits, where 2^N is the number of points of the selected modulation scheme (i.e., $N = 2$ for QPSK, $N = 4$ for 16-QAM, $N = 6$ for 64-QAM and $N = 8$ for 256-QAM).

DL-MAP generation and TAS functionality

The main purpose of the DL-MAP was to provide the receiver with detailed information regarding the subcarrier allocation utilized in each AMC-formatted OFDM symbol. A total of 2880 subcarriers are available in the two PUSC OFDM symbols comprising the DL-MAP. Additionally, it is mandatory to utilize the QPSK modulation in these OFDM symbols, according to the WiMAX standard. Consequently, 5760 bits can be disposed to describe the format of the closed-loop frame. Similarly to the previously described feedback-structure, the DL-MAP defines, for each 16-subcarrier group, the utilized transmit antenna, the destination user (i.e., given the point-to-point nature of the DL communication, a single fixed value is used) and the selected modulation scheme. The designed DL-MAP structure is shown in Figure 5.20. Considering that it is transmitted utilizing a STBC-based communication link, a basic Forward Error Correction - FEC - scheme - based on a (7,1) repetition code - was applied to ensure the reliable decoding of such vital subcarrier-allocation information. Moreover, the performance of the FEC scheme was verified by means of MATLAB/VHDL co-simulations utilizing realistic signal captures.

To compose the DL-MAP, the transmitter utilizes the received feedback. In case that no new feedback is received before it is required to generate the following frame, the last received feedback-information is utilized (or a predefined one for the first transmission). Finally, the dynamic allocation of the subcarriers is performed by forwarding each group of modulated symbols to the processing branch of the selected transmit antenna (i.e., as indicated by the feedback).

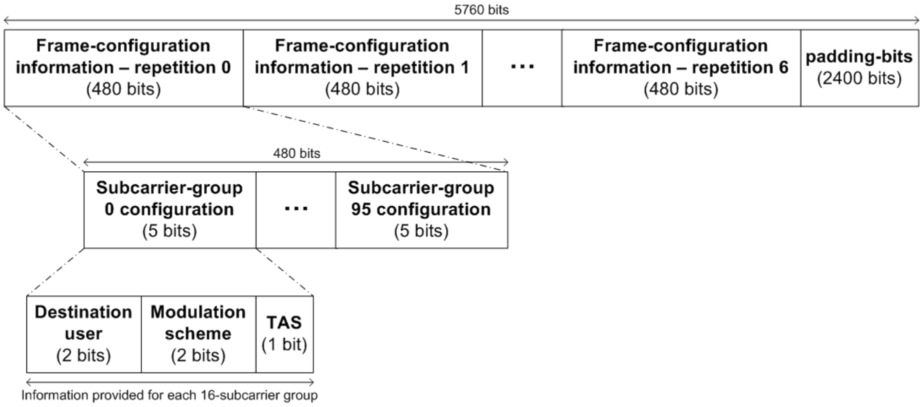

5760 bits
Frame-configuration information – repetition 0 (480 bits)
Frame-configuration information – repetition 1 (480 bits)
...
Frame-configuration information – repetition 6 (480 bits)
padding-bits (2400 bits)
480 bits
Subcarrier-group 0 configuration (5 bits)
...
Subcarrier-group 95 configuration (5 bits)
Destination user (2 bits)
Modulation scheme (2 bits)
TAS (1 bit)
Information provided for each 16-subcarrier group

Figure 5.20: Custom DL-MAP structure.

Multi-antenna closed-loop receiver

The baseband architecture of the closed-loop MIMO receiver is depicted in Figure 5.21. The DSP stages up to the FFT are the same ones designed for its STBC-based counterpart. Similarly, those operations related to the decomposition of the slots were extended to include the AMC permutation scheme. Hence, the main design efforts are concentrated on the channel estimation, symbol decoding, de-mapping and all those aspects related to the run-time adaptivity of the DSP processing blocks, as described in the following.

Pilot extraction and channel estimation

The pilot distribution utilized in the AMC slots for the two-antenna system can be observed in Figure 5.22. Exactly as in the PUSC permutation scheme, it is required to utilize two consecutive OFDM symbols in order to estimate the channel.

Exactly as in the open-loop STBC-based receiver, the arithmetic formulation employed to calculate the channel estimation is given by (5.17) and (5.18). Its efficient design and implementation is discussed in Section 5.3.2.

Joint STBD and MRC processing architecture

To recover the DL-MAP it is required to apply the exact STBD process described for the open-loop multi-antenna receiver. To decode the remaining OFDM symbols it is then applied an adapted version of the MRC architecture designed for the SIMO system. In the TAS scheme a 1x2 SIMO transmission is in fact applied, with the particularity that the transmit antenna dynamically varies. The information provided by the DL-MAP is utilized to configure this adaptive MRC processing. The TAS-adapted formulation of (5.13) can be expressed as follows:

$$\hat{d}_k = \frac{\sum_{i=1}^{n_R} S_i[k] \cdot \tilde{H}^*_{i,T}[k]}{\sum_{i=1}^{n_R} |\tilde{H}_{i,T}[k]|^2} \tag{5.21}$$

where $\tilde{H}_{i,T}[k]$ represents the estimate of the channel frequency response for the kth subcarrier from the selected transmit antenna (according to the TAS scheme), T, for each receive antenna i. The information regarding the transmit antenna that was utilized and, hence, which of the channel-estimation coef-

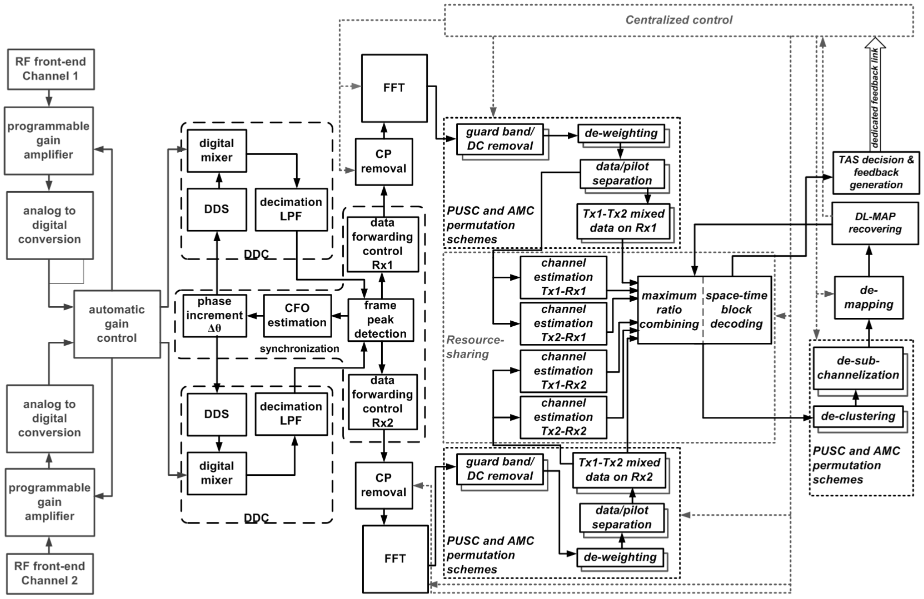
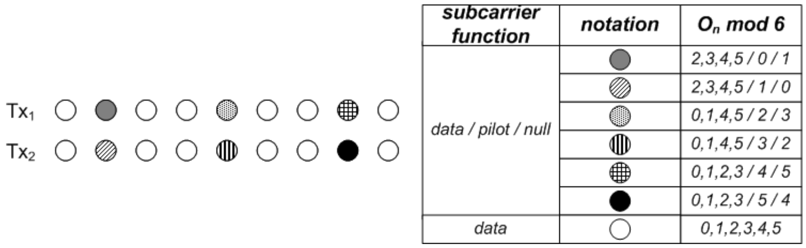


Figure 5.21: General architecture of the MIMO closed-loop receiver.



subcarrier function	notation	O_n mod 6
data / pilot / null	◉	2,3,4,5 / 0 / 1
	▨	2,3,4,5 / 1 / 0
	░	0,1,4,5 / 2 / 3
	▥	0,1,4,5 / 3 / 2
	▦	0,1,2,3 / 4 / 5
	●	0,1,2,3 / 5 / 4
data	○	0,1,2,3,4,5

Figure 5.22: Distribution of the pilot subcarriers in the two-antenna AMC configuration.

ficients are required, at each time instant, to compute (5.21) is obtained by decoding the DL-MAP (detailed below).

A joint design of the STBD and MRC schemes was conducted to enable a resource-efficient implementation, as described in Section 5.3.3.

Symbol de-mapping

The de-mapper stage could be programmed to recover the original bit-sequence in groups of N bits, according to the utilized modulation scheme (which is indicated in the DL-MAP).

DL-MAP recovery

Accounting for the applied (7,1) repetition code, and for the described DL-MAP structure (recall Figure 5.20), a 480-bit accumulator is utilized to add up each of the repetitions of the single-bits of the de-mapped PUSC sequence. By this way it is implicitly decided which information was originally transmitted (i.e., $if\ \sum_{i=0}^{6} bitSeq[k,i] > 3\ then\ original\_bitSeq[k] = 1\ otherwise$

$original\_bitSeq[k] = 0$, where i represents the seven repetitions).

TAS-decision and feedback generation

As previously described, the TAS-decision is performed in the receiver, which communicates the resulting subcarrier allocation for the following frame to the transmitter via a dedicated feedback link. Specifically, Algorithm 1 is applied to the last OFDM symbol of each frame (i.e., the TAS-decision metric is calculated using the last OFDM symbol of the received frames).

Algorithm 1 TAS-allocation algorithm.

```
  $f = 0;$
  $T_{0,0} = 2; T_{1,0} = 1; T_{2,0} = 2; ...; T_{95,0} = 2; T_{96,0} = 1;$
  for each received frame($f$) do
    $f = f + 1;$
    for each 16-subcarrier set ($G$) in the last OFDM symbol of the frame do
      $$T_{G,f+1} = \begin{cases} T_{G,f} & \text{when } SQI_{G,T} \leq SQI_{G,NT} \\ NT_{G,f} & \text{when } SQI_{G,T} > SQI_{G,NT} \end{cases} \quad (5.22)$$
    end for
  end for
```

In the algorithm f identifies each received frame (i.e., the predefined values used in the first transmission, where no feedback is available yet, are defined using $f = 0$), G identifies each group of 16-subcarrier within an OFDM symbol, $T_{G,f+1}$ is the transmit antenna selected for the Gth subcarrier group in the following frame, $T_{G,f}$ is the antenna selected to transmit the Gth subcarrier group, $NT_{G,f}$ is the transmit antenna in silence during the Gth subcarrier group and $SQI_{G,t}$ is the signal-quality indicator for the Gth subcarrier group for the transmit antenna t.

In order to simplify the overall implementation complexity, it was decided to extract the signal-quality indicator metrics directly from the MRC calculations. In more details, the TAS-decision metrics are calculated as indicated hereafter:

$$SQI_{G,T} = \sum_{k=1+(G-1)\cdot 16}^{G\cdot 16} |\tilde{H}_{1,T}[k]|^2 + |\tilde{H}_{2,T}[k]|^2 \tag{5.23}$$

$$SQI_{G,NT} = \sum_{k=1+(G-1)\cdot 16}^{G\cdot 16} |\tilde{H}_{1,NT}[k]|^2 + |\tilde{H}_{2,NT}[k]|^2 \tag{5.24}$$

During the MRC-processing of the last OFDM symbol in each frame, the four channel response estimates, $\tilde{H}_{i,j}[k]$, are forwarded to the TAS-decision and feedback generation block, which calculates (5.23) and (5.24). Then, Algorithm 1 is applied resulting in a feedback signal, which is as indicated in Figure 5.18. An illustrative example of the resulting subcarrier allocation is provided in Figure 5.23.

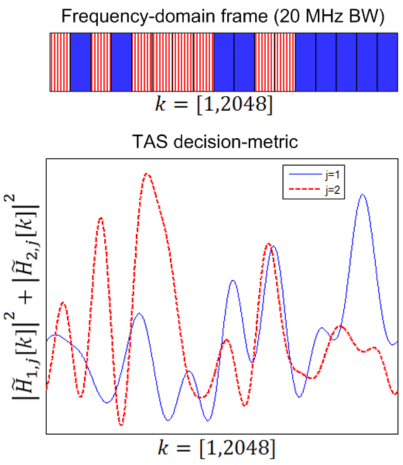


Figure 5.23: Illustrative example of dynamic subcarrier allocation according to the instantaneous values of the TAS-decision metrics.

5.3 Innovating RTL design techniques for the presented baseband schemes

The main contribution of the present case study is encountered in the featured RTL design techniques. The latter are tailored for bit-intensive OFDM-based PHY-layer implementations and, without loss of generality, they can be also applied in other high performance FPGA designs. The employed RTL design techniques resulted in a performance-efficient FPGA resource utilization.

Before entering in details, it is worth recalling that, according to the proposed development methodology, the previously introduced system models were manually translated to HDL code, taking into account the specifications of the target prototyping platform and the resources provided by the FPGA devices that it comprises. The overall design efficiency is increased by utilizing pre-verified FPGA designs; the Xilinx IP core library provides an efficient implementation of the common DSP functions (e.g., CORDIC, FFT, FIR filters) and fixed-point arithmetic operations (e.g., pipelined divider). Besides their underlying optimized low-level design, some important aspects of its RTL architecture are configurable (e.g., degree of parallelism of the internal calculations or the specific utilization of the DSP48 slices available in the target FPGA devices). The proper selection of these configuration parameters results essential to meet the system requirements, while attaining an optimum trade-off between resource-utilization and performance. The optimum values for such parameters were obtained by means of exhaustive MATLAB/VHDL co-simulations utilizing realistic signal data (i.e., acquired by using an off-line prototyping approach).

5.3.1 Synchronization

The synchronization is the heart of the DFE and an overall delicate and processing-consuming stage for almost all the high-performance wireless communication systems. In a real-life system, its implementation complexity scales with the

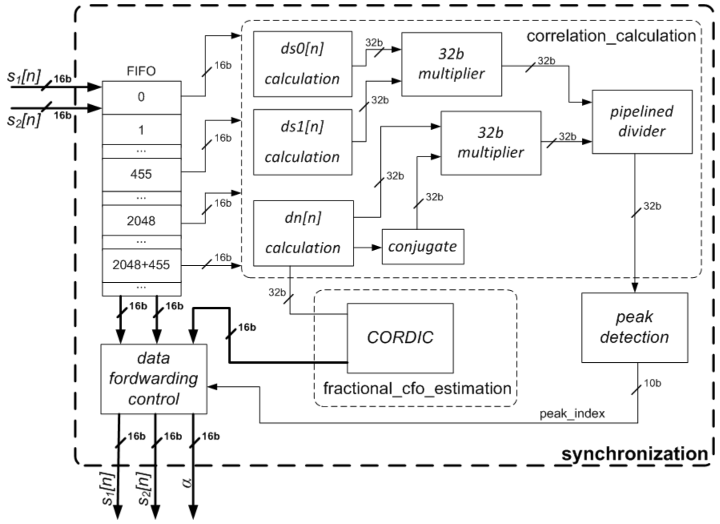


Figure 5.24: Architecture of the synchronization block.

size of the bandwidth. Moreover, the synchronization must efficiently address the effects of the channel and those of the signal impairments (e.g., hardware-originated).

The computational complexity of the synchronization can be reduced by optimizing certain algorithmic operations, minimizing likewise the arithmetic-implementation cost. Indicatively, due to the resource-demanding implementation of the cross-correlation defined by (5.12) for the two-antenna MIMO scheme, the following simplification in terms of complexity was applied:

$$|r_s[n]|^2 = \frac{|dn[n]|^2}{ds0[n] \cdot ds1[n]}, \tag{5.25}$$

where:

$$dn[n+1] = \begin{cases} dn[n] + \sum_{i=1}^{n_R} s_i^*[n+455] \cdot s_i[n+2048+455] & \text{if n} \leq 455 \\ dn[n] - \sum_{i=1}^{n_R} s_i^*[n] \cdot s_i[n+2048] & \\ \qquad + \sum_{i=1}^{n_R} s_i^*[n+455] \cdot s_i[n+2048+455] & \text{if n} > 455, \end{cases} \tag{5.26}$$

with $dn[0] = 0$. It should be noted that $ds0[n]$, $ds1[n]$ are calculated in a similar manner. With this optimization only four samples (per receive antenna) need to be introduced to the already calculated correlation, thus resulting in a reduced-complexity FPGA realization.

The general architecture of the proposed synchronization scheme is shown in Figure 5.24. Due to the stringent real-time constraints, a pipelined structure was developed for the calculation of the simplified cross-correlation. More specifically, accounting for the I/Q-composition of $s_i[n]$, the fixed-point arithmetic operations were decomposed into the basic arithmetic operation of the real and imaginary operands, as detailed in Figure 5.25. For the (real) division it was utilized a pipelined divider (efficiently implemented by a Xilinx IP core). As a result, a latency of 55 clock cycles is required to calculate (5.25). FIFO memories provided a latency-leveller temporary storage for the incoming signal.

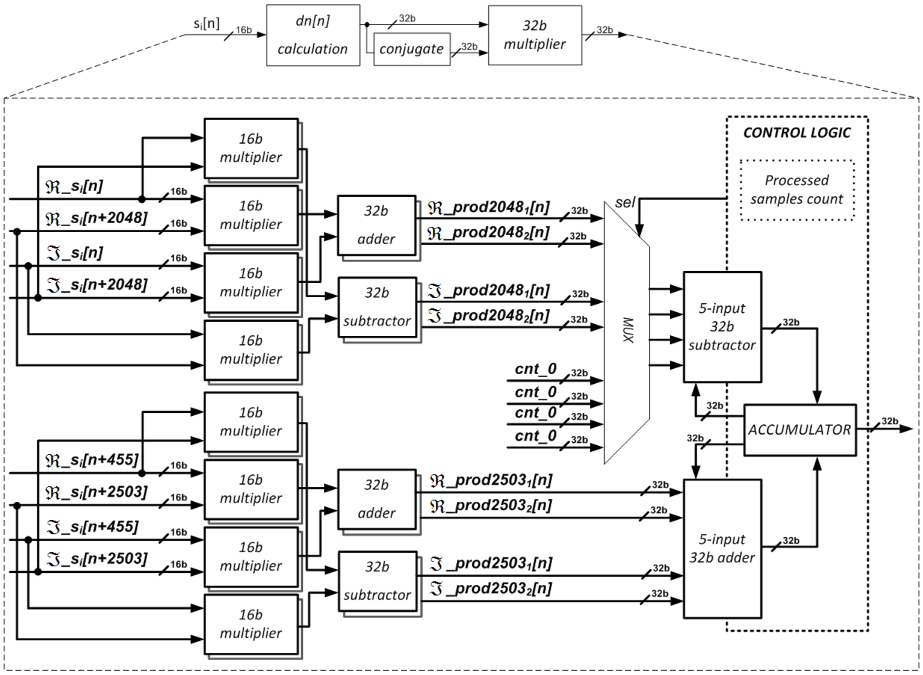
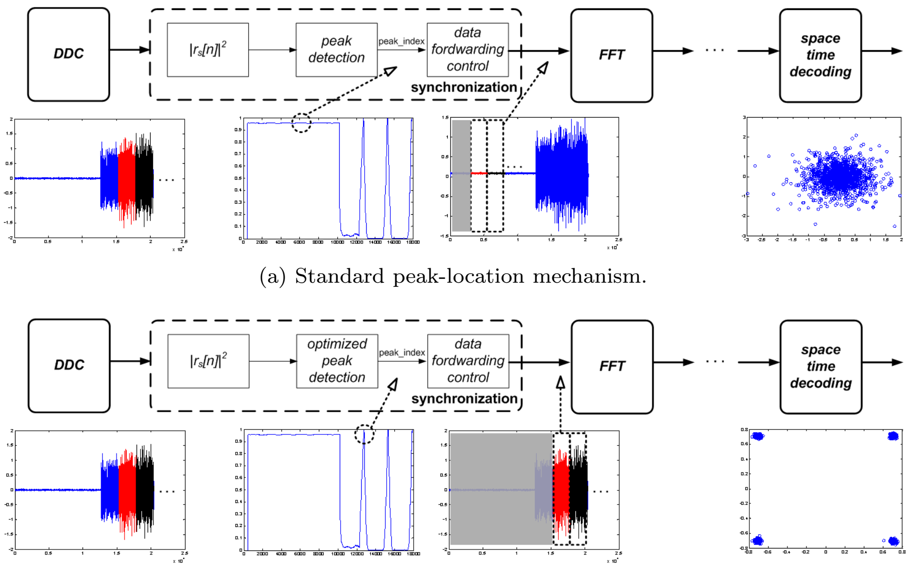


Figure 5.25: Detailed diagram of the pipelined implementation of the simplified cross-correlation.

A custom design for these memories, based on embedded memory blocks, allowed the retrieval of the four specific samples which enabled the calculation of the cross-correlation at each clock cycle. Additionally, once the location of the first sample of each OFDM symbol is determined by the data-forwarding control logic, the window of subcarriers comprising the entire frame is synchronously forwarded. Finally, it must be noted that the internally utilized registers are reset at the beginning of a newly detected frame, in order to avoid problems with the accumulated arithmetic-error (i.e., due to the finite precision of the utilized fixed-point representation).

The detection of the correlation peak is a critical part of the synchronization algorithm because it is error-prone under the presence of spurious signals. The regular operation of the GEDOMIS testbed introduces a parasitic sinusoid to the received baseband signal because of the digital mixing of the unwanted residual carrier at the center of the useful signal spectrum (i.e., LO coupling at the transmitter) happening in the DDS stage (as indicated by the MIMO signal model (5.15)). The presence of this sinusoid during the silence period can result in erroneous performance of the aforementioned correlator, which may produce false peaks (i.e., not corresponding to the detection of the CP) and consequently misplace the window of samples forwarded to the FFT processing block. Failure to prevent an erroneous OFDM symbol synchronization may render the system unusable. Thus, the peak detection algorithm, which is usually based on a triggering threshold and the selection of the maximum value in a window, must be optimized to recognize the legitimate peaks. When the trigger issues a correlation value above the threshold, the shape of the correla-



(a) Standard peak-location mechanism.

(b) Optimized peak-location mechanism.

Figure 5.26: Effects of a DC level on the received signal.

tion curve determines whether the detected peak indicates the beginning of an OFDM symbol or a false peak occurring during a silence period. The correlation curve tends to have high-values and nearly no variations during the silence periods (Figure 5.26a), whereas it features high values only during the processing of the CP preceding each OFDM symbol (Figure 5.26b). After exhaustive MATLAB simulations (using experimental data obtained with GEDOMIS), it was observed that the correlation peaks corresponding to the detection of a CP were always located in a window of 300 samples, after the first correlation value above the defined threshold (which was set at 0.5). A normalized sum of the defined sample-window was then used to decide about the legitimacy of the peak under scrutiny. Specifically, the following equation was implemented:

$$NS_m = \frac{\sum_{l=m}^{m+299} r_s[l]}{r_s[l_{peak}]}, \tag{5.27}$$

where $r_s[m]$ represents the first correlation value above the defined triggering-threshold and $r_s[l_{peak}]$ is the peak-value of the inspected 300-sample window (i.e., $l_{peak} = \arg\ \max_l r_s[l]$). A value close to the number of added samples (i.e., $NS_m > 270$) indicates a false peak, otherwise the peak-value indicates the detection of the CP.

5.3.2 Channel estimation architecture

In a MIMO scheme the receiver needs to estimate the channel of each transmit-receive antenna pair, which obviously implies an intensive computation. When this is combined with a wide signal bandwidth and a complicated frame structure, such as the one featured in the mobile WiMAX standard, big chunks of data need to be processed accounting for the distribution of the pilot subcarriers

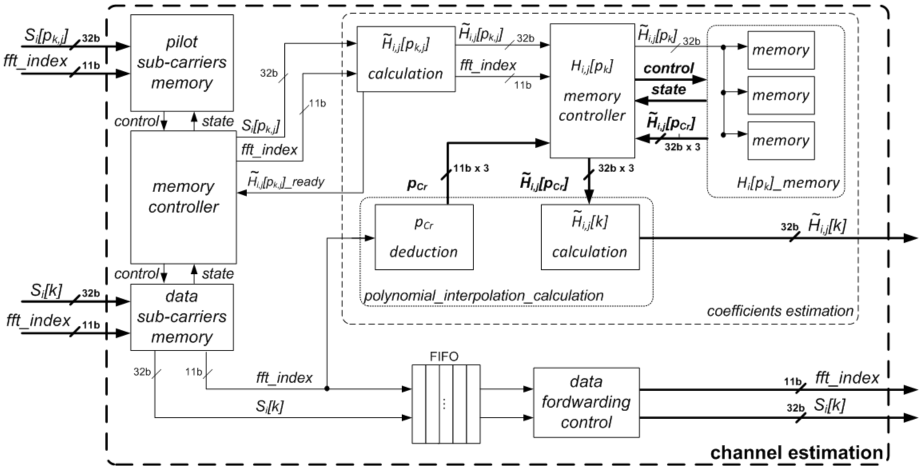


Figure 5.27: Architecture of the channel estimation block.

among various OFDM symbols. The real-time implementation of the systems presented in this chapter required a speed-optimized pipelined architecture, in order to cope with the stringent performance prerequisites.

Figure 5.27 presents the general processing architecture of the channel estimation utilized for the 2x2 MIMO open-loop and closed-loop receivers. The two flows resulting from the data/pilot separation are stored in two separated memories (i.e., one for the pilot subcarriers and another for the those I/Q values containing the user data). This storage is necessary considering the utilized multi-antenna PUSC and AMC structures (i.e., the channel estimation is only commencing when all the pilot carriers of two consecutive OFDM symbols are received; otherwise the separation among the pilots utilized in the polynomial interpolation would be excessively high, resulting in a decrease of accuracy for the channel estimation).

The channel frequency response at the pilot tones is estimated according to (5.17). A carefully designed in-block memory structure is required to store the calculated coefficients, until the channel estimates are computed at the positions of all the pilot subcarriers, $\widetilde{H}_{i,j}[p_{k,j}]$. The calculation of the channel frequency response at each data subcarrier needs to utilize three neighbouring pilot tones. Hence, three memory elements capable of storing 1728x32 bits were defined. Each estimated coefficient has a 32 bit length; the AMC OFDM symbols comprise 1728 active subcarriers (i.e, $n_U = 1727$). The calculated $\widetilde{H}_{i,j}[p_{k,j}]$ is stored according to the FFT output-index of $p_{k,j}$ (i.e., location of the pilot within the OFDM symbol, once the DC and guard-bands are removed). The resulting pilot-coefficient values are replicated in the three memory elements. While the memory-utilization is not optimum (i.e., only a portion of its maximum storage capacity is utilized), the two previous features allowed to recover any three-pilot set with a minimum latency of one clock cycle, while avoiding extra computational complexity in the memory-management plane. The described design provides the pipelined architecture that calculates $\widetilde{H}_{i,j}[k]$, based on the polynomial interpolation defined in (5.18). Finally, a FIFO memory is used to compensate the latency introduced by the calculations and supply an

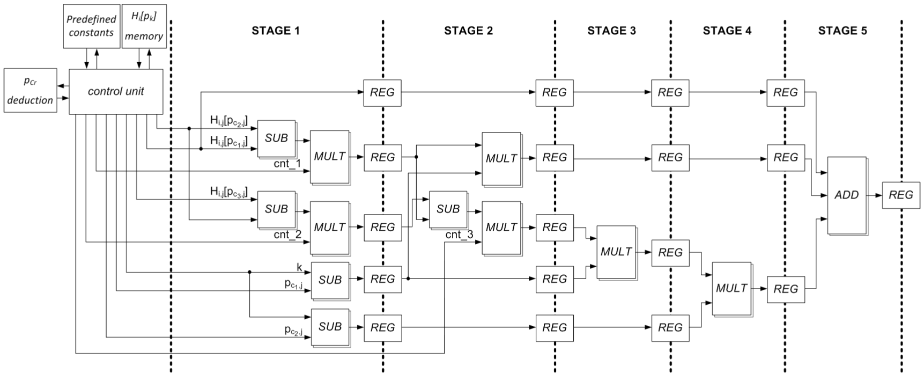


Figure 5.28: Detail of the pipelined architecture implementing the second order polynomial interpolation.

aligned output.

More specifically a 5-stage pipeline was designed, based on the following decomposition of (5.18):

$$\tilde{H}_{i,j}[k] = Op1+((Op2_1\cdot cnt\_1)\cdot Op2_2)+(((Op3_1\cdot cnt\_2)-Op2_1)\cdot cnt\_3)\cdot Op2_2\cdot Op3_2, \tag{5.28}$$

where $cnt\_1 = \frac{1}{p_{c_2,j}-p_{c_1,j}}$, $cnt\_2 = \frac{1}{p_{c_3,j}-p_{c_2,j}}$ and $cnt\_3 = \frac{1}{p_{c_3,j}-p_{c_1,j}}$ can be calculated a priori for any possible combination of neighbour pilots (i.e., predefined constants that can be retrieved when required). In other words, the utilization of these constants provides a division-free implementation of (5.18). Indicatively, $Op2_1 \cdot cnt\_1$ results in an efficient utilization of DSP48 slices to implement $\frac{\tilde{H}_{i,j}[p_{c_2,j}]-\tilde{H}_{i,j}[p_{c_1,j}]}{p_{c_2,j}-p_{c_1,j}}$. Figure 5.28 gives an insight of the mentioned pipelined architecture. It is important to underline that while $H_{i,j}$ is complex-valued, the other operands involved in the defined arithmetic calculations are real-valued. Consequently, duplicated multiplication, subtraction and addition blocks are required for $\tilde{H}_{i,j}$.

It must be noted that the data subcarriers, after the FFT calculation, do not maintain their sequential order. Therefore it is necessary an algorithm to calculate the indexes of the three neighbour pilot subcarriers, p_{c_r}, accounting for the position of the actual subcarrier at the output of the FFT. More specifically, the position of any subcarrier needs to be expressed according to the utilized permutation scheme. For the PUSC scheme, it can be easily expressed as:

$$k = 14 \cdot c + p, \tag{5.29}$$

where $c \in \{0, ..., 119\}$ identifies each of the 120 PUSC clusters composing the OFDM symbol and $p \in \{0, ..., 13\}$ indicates the precise position of the subcarrier within the cluster. Similarly, for AMC-formatted symbols, $c \in \{0, ..., 191\}$ identifies each BIN, with $p \in \{0, ..., 8\}$. The associated values of c and p are provided by the preceding data/pilot separation block; further details regarding the calculation of c and p are provided in Section 5.3.5.

To illustrate the precise algorithm utilized to select the three neighbouring pilots for each data subcarrier, hereafter it is explained an example for those PUSC-formatted symbols where $O_n \mod 4 \in \{0, 1\}$. Taking into account the precise pilot distribution (Figure 5.13) and the pilot-transmission scheme (i.e., one antenna transmits one pilot subcarrier while the other remains quiet), the two pilots within each cluster, c, are always found in $p = 4$ and $p = 8$. This leads to three possible sets of neighbouring pilots, as indicated in Figure 5.29[3]. Based on (5.29), the relative positions of the neighbour pilots corresponding to the kth carrier can be calculated easily, as indicated below:

$$current\_cluster\_first\_pilot = \begin{cases} k+4-p & \text{if k <840} \\ k+4-p+1 & \text{if k>840} \end{cases} \tag{5.30}$$

$$current\_cluster\_second\_pilot = \begin{cases} k+8-p & \text{if k <840} \\ k+8-p+1 & \text{otherwise} \end{cases} \tag{5.31}$$

$$previous\_cluster\_second\_pilot = \begin{cases} k-6-p & \text{if k <840} \\ k-6-p+1 & \text{otherwise} \end{cases} \tag{5.32}$$

$$following\_cluster\_first\_pilot = \begin{cases} k+18-p & \text{if k <840} \\ k+18-p+1 & \text{if k>840} \end{cases} \tag{5.33}$$

$$following\_cluster\_second\_pilot = \begin{cases} k+22-p & \text{if k <840} \\ k+22-p+1 & \text{otherwise,} \end{cases} \tag{5.34}$$

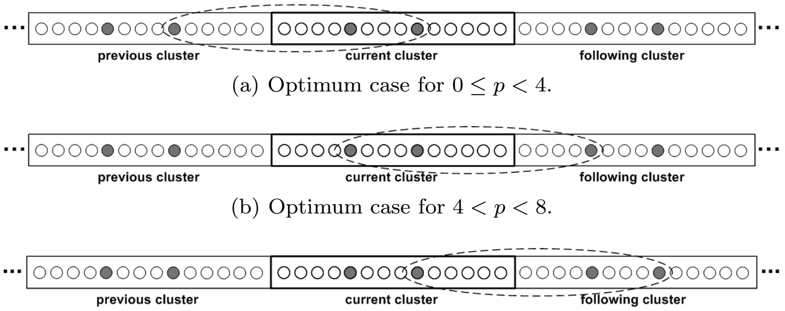
where $k = 480$ is the position of the DC in a PUSC-formatted OFDM symbol after removing the guard-bands (i.e., the DC was removed before the channel estimation, but it was present during the propagation of the signal over the channel). The same scheme is repeated for the remaining PUSC and AMC pilot distributions. Finally, the three pilots can be readily recovered from the dedicated pilot-coefficients memories, considering that their indexes are already known.

5.3.3 Joint design of the STBC and MRC techniques

The closed-loop system configuration required the co-existence of two different transmission techniques and an adaptive response according to the instantaneous channel conditions. This increased the required processing resources and overall architectural complexity. Hence, to attain a hardware-efficient implementation, an accurate RTL design is needed, maximizing the resource-reutilization among the two transmission schemes, without compromising the expected performance, latencies and overall efficient design of each separate configuration. This is made challenging considering that each transmission scheme utilizes different permutations.

At the receiver, the open-loop configuration is based on a STBC scheme, whereas the closed-loop one uses a TAS scheme to transmit the AMC-formatted

[3] The optimum 3-pilot set for each data subcarrier within the cluster was defined through extensive simulations of the corresponding MATLAB model.



(a) Optimum case for $0 \leq p < 4$.

(b) Optimum case for $4 < p < 8$.

(c) Optimum case for $8 < p \leq 13$.

Figure 5.29: Possible grouping of neighbouring pilots.

data bursts. In order to decode the data symbols at the closed-loop receiver, a MRC technique was applied. The STBD formulation - defined in (5.19) and (5.20) - and its MRC counterpart - defined in (5.21) - feature a high degree of similarity. This resemblance was exploited to optimize the FPGA-resource usage, by providing a joint RTL design of the STBD and MRC schemes.

As in the previously described cases, all the fixed-point calculations involving complex numbers were decomposed into the real and imaginary operands, resulting in two concurrently operating processing branches. An important aspect of the design, resides in the sign-change of the imaginary part due to the conjugation of the complex values, which can be expressed as:

$$\Re(S_i[k] \cdot \tilde{H}^*_{i,j}[k]) = \Re(S_i[k]) \cdot \Re(\tilde{H}_{i,j}[k]) + \Im(S_i[k]) \cdot \Im(\tilde{H}_{i,j}[k]) \tag{5.35}$$

$$\Im(S_i[k] \cdot \tilde{H}^*_{i,j}[k]) = \Im(S_i[k]) \cdot \Re(\tilde{H}_{i,j}[k]) - \Re(S_i[k]) \cdot \Im(\tilde{H}_{i,j}[k]) \tag{5.36}$$

$$\Re(S^*_i[k] \cdot \tilde{H}_{i,j}[k]) = \Re(S_i[k]) \cdot \Re(\tilde{H}_{i,j}[k]) + \Im(S_i[k]) \cdot \Im(\tilde{H}_{i,j}[k]) \tag{5.37}$$

$$\Im(S^*_i[k] \cdot \tilde{H}_{i,j}[k]) = -\Im(S_i[k]) \cdot \Re(\tilde{H}_{i,j}[k]) + \Re(S_i[k]) \cdot \Im(\tilde{H}_{i,j}[k]) \tag{5.38}$$

Additionally, the design of this processing block takes into account the timing characteristics of the preceding channel estimation block, which is providing the required input data and control signals (recall Figure 5.21). As already mentioned, the pilot tones are distributed in groups of two consecutive OFDM symbols, which implies that the channel estimation is performed per OFDM symbol-pairs; hence, the STBD and MRC operations are inheriting this functionality. This has as a consequence, that equations (5.35) and (5.36) are always applied to the data symbols of the first OFDM symbol of the pair (i.e., O_n), while the data symbols of the second OFDM symbol (i.e., O_{n+1}) are processed applying the equations (5.37) and (5.38). Additionally, it can be observed:

$$\Re(S_i[k] \cdot \tilde{H}^*_{i,j}[k]) = \Re(S^*_i[k] \cdot \tilde{H}_{i,j}[k]) \tag{5.39}$$

$$\Im(S_i[k] \cdot \tilde{H}^*_{i,j}[k]) = -\Im(S^*_i[k] \cdot \tilde{H}_{i,j}[k]) \tag{5.40}$$

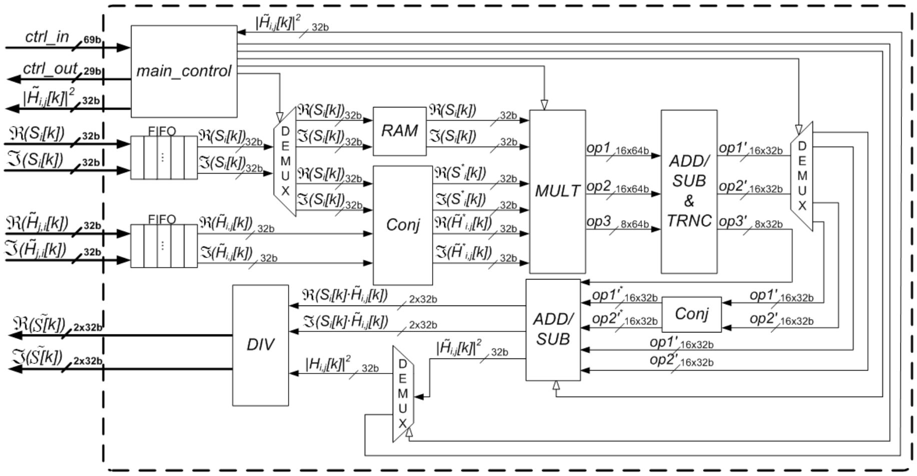


Figure 5.30: High-performance pipelined architecture designed to optimally implement the space-time decoding and MRC operations at the receiver.

Hence, based on (5.40), it can be noticed that the same logic used to calculate (5.19) and (5.20) can be reused to calculate (5.21), adjusting the sign of (5.38) when processing the second OFDM symbol of each pair in the AMC-formatted zone of the frame. Likewise, the proposed RTL architecture efficiently accommodates both transmission schemes.

Figure 5.30 shows the high-performance 5-stage pipeline structure that was designed for the joint processing of the two OFDM symbols. In the first stage the multiplication of the different real and imaginary operands is calculated. All the required calculations in equations (5.35), (5.36), (5.37) and (5.38) are completed in the second stage. The third stage is providing the addition and subtraction operations required to calculate the numerator and denominator of equations (5.19), (5.20) and (5.21). At this point, the main controller of the block is taking into account (5.40) in order to correct (when required) the sign of the operands. The fourth stage is dedicated to fixed-point divisions (i.e., this last division is simplified due to the fact that the denominator is a real operand), and finally the fifth stage provides the essential control-signalling together with the decoded symbols. Additionally, the internal fixed-point arithmetic calculations satisfied an optimum trade-off between computational complexity and result precision. Indeed, the co-simulation of the VHDL and MATLAB descriptions indicated that 64 bits could be utilized for the internal arithmetic calculations, attaining a fixed-point implementation-loss at the order of 10^{-8}.

Figure 5.31 presents the operation of the designed pipeline and the associated latencies. As it can be observed, a total of 75 clock cycles are employed for the calculations; initially it is required to store one OFDM symbol per antenna before applying on-the-fly the STBD or MRC operations during the reception of the second OFDM symbol of each pair. In other words, two data symbols are estimated at each operation of the block, with an initial latency equal to the length of an OFDM symbol. The resulting time-budget between each pair of OFDM symbols at the output of the channel estimation satisfies the real-time timing requirements of the DL-MAP decoding. The information included in

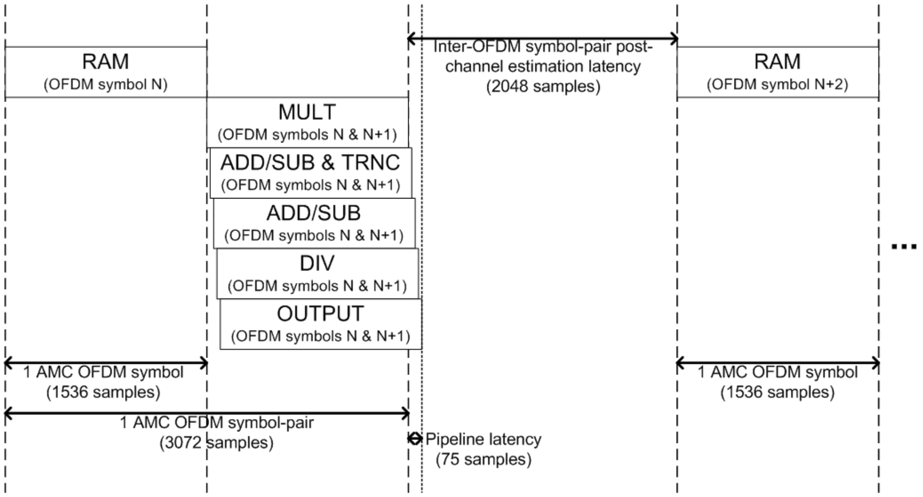


Figure 5.31: Low latency 5-stage pipelined design (STBD and MRC).

the two initial PUSC-formatted OFDM symbols has to be decoded and timely processed by the main controller of the closed-loop receiver, in order to configure the MRC processing before the first pair of AMC-formatted OFDM symbols is ready to be decoded; this prevents the buffering of the complete incoming frame. It is also worth mentioning that there is no need to store the channel coefficients for the first OFDM symbol of the pair, because they are the same ones with the posterior one. Finally, the resulting symbols enter the de-permutation and de-clustering processing stages, while at the same time the calculated $SQI_{G,T}$ and $SQI_{G,NT}$ values are processed by the TAS-decision-making and feedback-generation processing block.

Table 5.3 compares the synthesis results of the STBD and MRC blocks when synthesized as independent processing blocks (i.e., without sharing functional processing components and calculations), with the synthesis results of the joint design presented herein. A slice-saving of 18% is achieved when targeting a Xilinx Virtex-4 LX160 FPGA. This is important considering that an optimized RTL-design strategy not only could save FPGA resources, but also it could allow the use of low-cost FPGA devices with all the advantages that this may imply. Moreover, this type of optimizations play a key role when designing the PHY-layer of real-time systems that have stringent FPGA area utilization constraints.

XC4VLX160	Slices	DSP48	RAMB16s
Non-optimized design	37269 (55%)	96 (100)%	12 (4%)
Optimized design	25068 (37%)	96 (100)%	12 (4%)

Table 5.3: Synthesis results showing the FPGA-resource savings of the designed RTL-architecture.

5.3.4 Adaptive memory structure

The mobile WiMAX standard is defining a flexible frame format serving the needs of multi-user cellular systems. Different service requirements need to be satisfied for each user, accounting for a number of different operating scenarios. For this reason the frames need to be dynamically formatted. For instance, each subcarrier permutation scheme is designed to serve a given communication technique (e.g., the AMC scheme results convenient for a closed-loop configuration with dynamic subcarrier allocation). The flexible frame construction supports any combination of the proposed permutation schemes.

From a design point of view, the above detailed flexibility considerably increases the complexity of the control and memory planes. More specifically, the precise zones composing each received frame define the subcarrier permutation operations to be applied, which involve a variable number of OFDM symbols. Thus, the memory accesses are based on groups of subcarriers that feature variable lengths and, thus, incur different latencies (i.e., for the write and read operations of consecutive OFDM symbols). For the reason mentioned before, it is essential to provide a high-throughput architecture based on an intelligent utilization of the available embedded memory-blocks.

Baseline design: single-antenna receiver

The baseline adaptive memory architecture features a performance-efficient design that enables the reception of frames which are able to include any possible combination of permutation-schemes. This complex memory architecture is the heart of the de-subchannelization processing stage of the receiver (recall Figure 5.6), whose role is to decompose the slots in order to retrieve the original sequence of the modulated symbols.

RTL-aware MATLAB modelling

The MATLAB model of the SISO system is supporting all the permutation schemes defined by the mobile WiMAX standard. Hence, following the proposed methodology, a RTL-aware model of the receiver was first designed, including a high-level description of the adaptive memory structure. In more detail, the adaptive storage architecture is composed by various memory-elements, each capable of storing one OFDM symbol. The worst-case size is provided by both FUSC and AMC permutation schemes, which include 1536 subcarriers (also, recall that at the output of the channel estimation the modulated symbols are represented using 32-bit I/Q values). Six of those memory blocks are required accounting for the case demanding the highest storage-capacity; the latter uses a frame that is formatted with 2x3 AMC OFDM symbols, where each slot includes three symbols. In this case, it is required to store the set of the three incoming OFDM symbols while the previous is being read.

The HLPL-based model provided an estimation of the required memory resources, but it also allowed the rapid functional evaluation of the proposed memory-architecture, as shown in Figure 5.33 (i.e., assuming the reception of a multi-user frame formatted according to the structure presented in Figure 5.32).

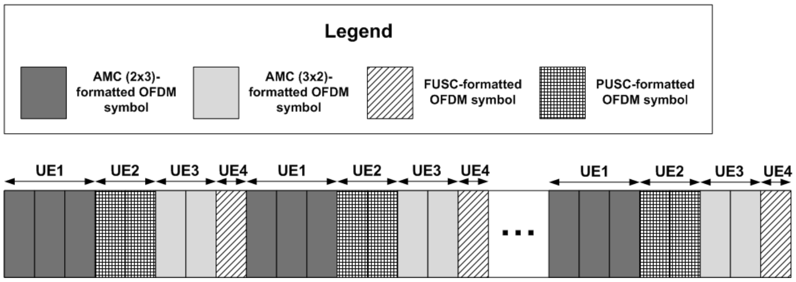
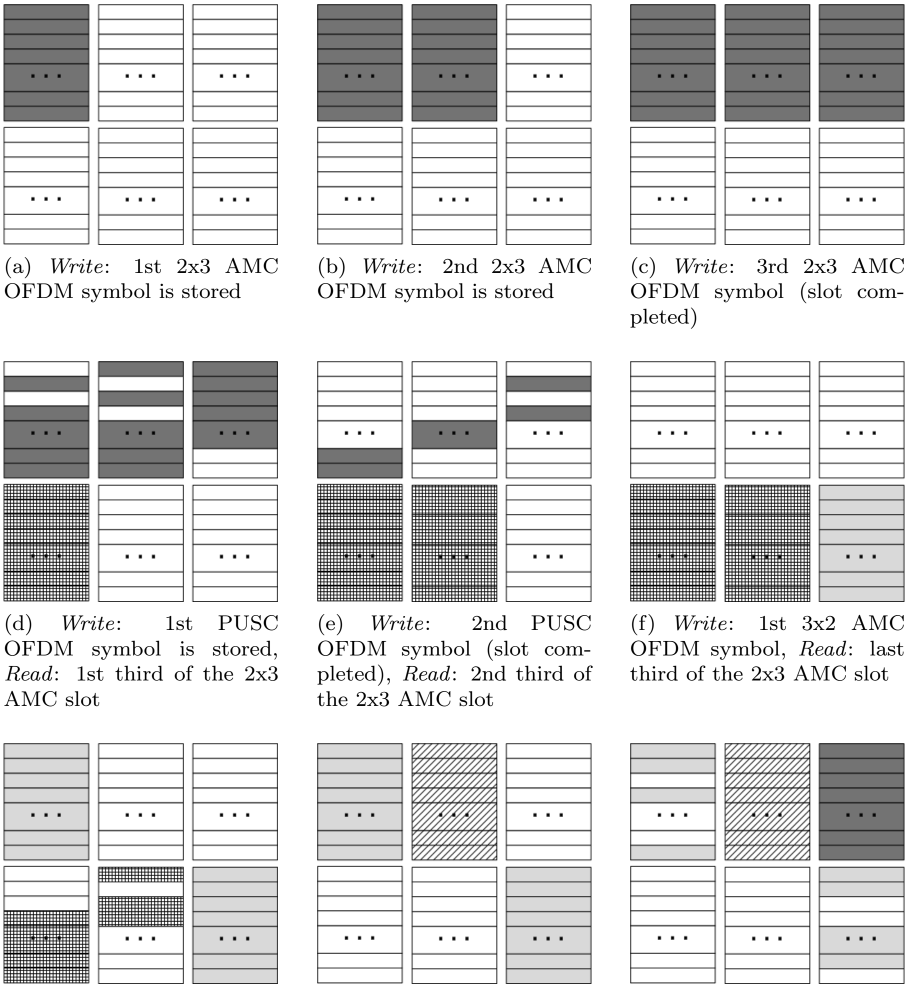

Legend
AMC (2x3)-formatted OFDM symbol
AMC (3x2)-formatted OFDM symbol
FUSC-formatted OFDM symbol
PUSC-formatted OFDM symbol
UE1
UE2
UE3
UE4
UE1
UE2
UE3
UE4
...
UE1
UE2
UE3
UE4

Figure 5.32: Format of the considered frame during the MATLAB simulations.



(a) *Write*: 1st 2x3 AMC OFDM symbol is stored

(b) *Write*: 2nd 2x3 AMC OFDM symbol is stored

(c) *Write*: 3rd 2x3 AMC OFDM symbol (slot completed)

(d) *Write*: 1st PUSC OFDM symbol is stored, *Read*: 1st third of the 2x3 AMC slot

(e) *Write*: 2nd PUSC OFDM symbol (slot completed), *Read*: 2nd third of the 2x3 AMC slot

(f) *Write*: 1st 3x2 AMC OFDM symbol, *Read*: last third of the 2x3 AMC slot

(g) *Write*: 2nd 3x3 AMC OFDM symbol is stored (slot completed), *Read*: 1st half of the PUSC slot

(h) *Write*: FUSC OFDM symbol (slot completed), *Read*: last half of the PUSC slot

(i) *Write*: 1st 2x3 AMC OFDM symbol (cycle restart), *Read*: 1st half of the 3x2 AMC slot

Figure 5.33: Illustrative operation case of the proposed adaptive memory structure.

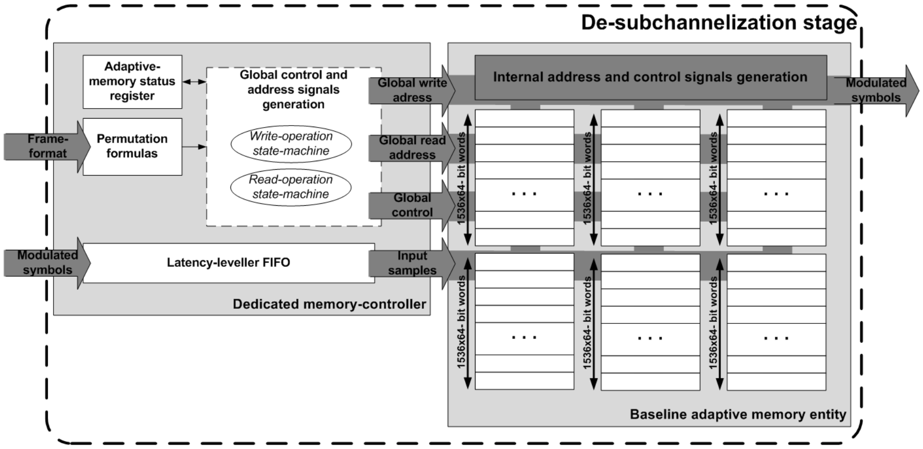

De-subchannelization stage
Adaptive-memory status register
Permutation formulas
Global control and address signals generation
Write-operation state-machine
Read-operation state-machine
Frame-format
Modulated symbols
Latency-leveller FIFO
Dedicated memory-controller
Global write adress
Global read address
Global control
Input samples
Internal address and control signals generation
Modulated symbols
1536x64- bit words
. . .
Baseline adaptive memory entity

Figure 5.34: Block-diagram of the baseline adaptive memory structure design.

HDL-based design

The RTL design goals were to provide a resource-efficient FPGA implementation with minimized latency. This is a complicated task considering the slot-based memory-access and the high-throughput data-path requirements of the proposed system. In more details, the real-time implementation utilizes a 20 MHz bandwidth, which results in a peak throughput of 179.2 MBytes/s for the inter-FPGA communications (accounting for the baseband sampling clock of 22.4 MHz and complex samples utilizing 32-bits for the real and imaginary components, at the output of the channel estimation stage).

The general RTL design of the baseline adaptive memory architecture is depicted in Figure 5.34. The adaptive memory entity groups the minimum required number of embedded memory elements, according to the de-subchannelization storage requirements. By abstracting away the internal memory block structure, it is enabled a flexible memory access and usage. In other words, the adjacent processing stages do not need to be aware of the low-level memory access and control requirements. Likewise, a seamless memory access is provided, since the de-clustering stage provides the complex samples jointly with the index resulting from its operation (the index indicates their position within the actual OFDM symbol); similarly, the de-mapping stage receives bursts of complex samples. A complex dedicated controller manages the synchronous memory accesses, according to the dynamic needs of the adjacent processing stages. The controller provides simultaneous write and read operations, which is a strict prerequisite in order to meet the performance requirements and manage the variable length of the sets of incoming subcarriers.

In order to implement the adaptive memory structure, the FPGA memory primitives were manually instantiated. The target FPGA devices (i.e., Xilinx Virtex-4 family) includes embedded RAM-blocks capable of storing 16 kbits of data, which can be utilized to construct single or double-port synchronous memories. Furthermore, different primitives are defined allowing to flexibly instantiate the embedded RAM-blocks in the HDL-code to efficiently create the required memory structure (e.g., allowing different data-word sizes)[4].

[4] A graphical interface is provided by Xilinx to automate the HDL-design of memory blocks. Nevertheless, a manual coding allows to fully define their low-level control logic, enabling a resource-efficient design, optionally including additional subcarrier operations that augment the memory-accesses (as it is detailed below).

Following the storage requirements defined by the RTL-aware MATLAB model, six 32x512-bit embedded RAM blocks were utilized to implement each of the six memory elements composing the adaptive memory entity; a total of 36 RAM instances are required to meet the storage requirements for the worst-case scenario (6x1536x64 bits). The detailed design can be observed in Figure 5.35. Half of the embedded RAM blocks are used to store the 32-bit real samples and the other half the imaginary values. Common low-level address and control signalling (i.e., RAM block level) is shared for both I/Q storage-branches, providing a simplified and efficient implementation of the internal control plane. The latter converts the general read and write addresses provided by the dedicated controller (respectively, *GRAddr* and *GWAddr* in Figure 5.35) to the required low-level addresses, by utilizing the classic addressing scheme based on N-to-2^N decoders and multiplexers. The three Most Significant Bits (MSBs) of the general address indicate which of the six memory elements is targeted, the next two bits indicate the specific I/Q RAM blocks (within the memory element) and the remaining Least Significant Bits (LSBs) provide the specific 32-bit word being accessed (within the 16-kbit RAM block). The signals controlling the specific type of access to the identified RAM-position (i.e., read or write) are also inferred by the decoder outputs. Finally, the operation of the internal control plane is driven by the general memory access requests, which are generated by the dedicated memory controller (ensuring a collision-free operation).

It is the dedicated memory controller the one providing the adaptivity to the operation of the designed storage architecture. First, simultaneous read and write operations are executed according to the specific slot-formatting of the current frame (recall Figure 5.33), providing minimized-latency and RAM-block usage. Hence, a dedicated register is used to control the status of the adaptive memory element (i.e., its contents). This register is updated every time that one OFDM symbol is read or written. Moreover, the read and write operations are optimized by implementing the required subcarrier permutation operations (i.e., decompose the slots to retrieve the original sequence of mapped symbols) with the minimum possible latency, while accounting for the variable length of the subcarrier-sets. Precisely, the design is based on the following:

(a) The subcarriers are written to the memory in sequential order (i.e., as received). A dedicated state-machine utilizes the status register to generate the precise number of required write requests and its associated global write addresses (i.e., recall that the memory accesses are slot-based).

(b) The subcarrier retrieval follows the ordering dictated by the scrambling-formulas corresponding to each permutation scheme. As a result of the previous, the specifically required global read addresses are generated accounting for the contents of the status register.

(c) A latency-leveller FIFO element is utilized to align the samples with the generated address and control signals.

High-throughput latency-optimized 2x2 MIMO transmitter

As seen in the single-antenna receiver, the flexible frame-formatting results in a de-subchannelization stage with high-throughput storage requirements. On the

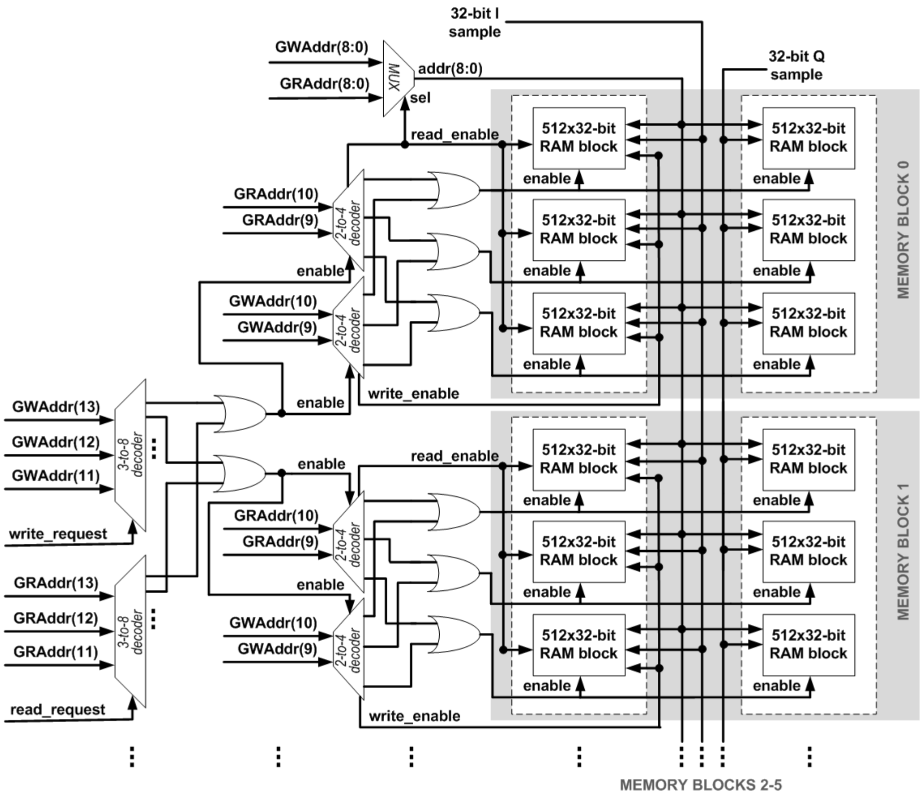


Figure 5.35: Detailed macro/gate-level design of the baseline adaptive memory structure.

contrary, in a mobile WiMAX transmitter those blocks implementing standard-related operations play a more crucial role, since the modulated symbols need to be permuted, inverted and grouped in diverse logical structures throughout the processing stages, before the IFFT is applied to the resulting OFDM symbols. In other words, at each processing stage of the transmitter, it is required a pipelined memory-structure with high-throughput to store large data-sets. Furthermore, the length of these sequences varies between the different stages of the baseband processing chain (e.g., additional non-data subcarriers are inserted or different permutation schemes are applied).

The previously presented baseline adaptive architecture constitutes a solid starting-point to implement the intermediate storage required in most of the processing stages of the transmitter. This baseline design was reused and extended to serve the needs of the maximum possible number of standard-related operations with the minimum required memory-resources and operation latency, by including the following features:

1. Additional logic was included to enable the concurrent execution of complementary baseband operations, while the storage and retrieval of the data sequences takes place. This parallelization prevented top-up latencies:

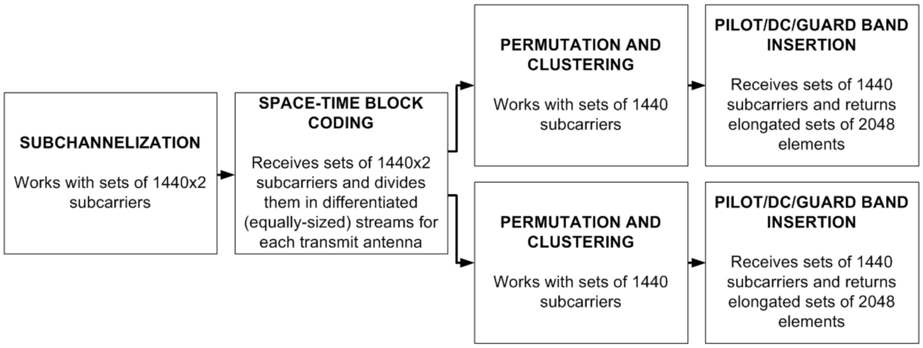

SUBCHANNELIZATION
Works with sets of 1440x2 subcarriers
SPACE-TIME BLOCK CODING
Receives sets of 1440x2 subcarriers and divides them in differentiated (equally-sized) streams for each transmit antenna
PERMUTATION AND CLUSTERING
Works with sets of 1440 subcarriers
PILOT/DC/GUARD BAND INSERTION
Receives sets of 1440 subcarriers and returns elongated sets of 2048 elements
PERMUTATION AND CLUSTERING
Works with sets of 1440 subcarriers
PILOT/DC/GUARD BAND INSERTION
Receives sets of 1440 subcarriers and returns elongated sets of 2048 elements

Figure 5.36: General outline of the memory-plane designed for the 2x2 MIMO STBC-based transmitter.

- If the operation is performed on a write memory-access, the additional logic is inserted in the dedicated memory controller.
- For read memory-accesses, it is the memory entity the one hosting the extra functionality.

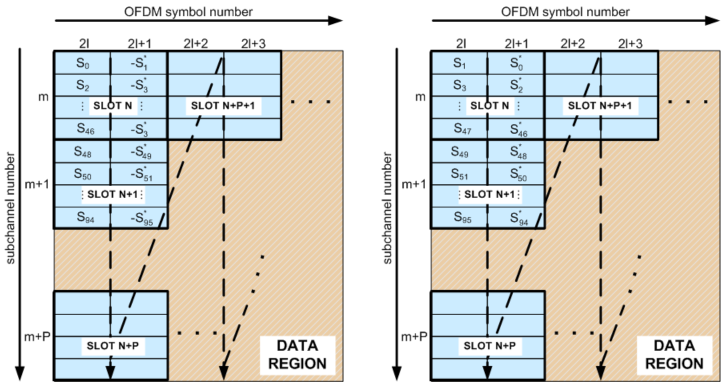
2. Both the write and read accesses are utilized (when required) to perform scrambling operations, minimizing the overall latency of the transmitter. Moreover, the overall capacities of the dedicated memory controllers have been extended in order to satisfy the increased adaptivity between processing stages.

In order to unveil the details of the incremental development flow, this section describes the extended design of the memory-plane for the 2x2 MIMO open-loop configuration of the mobile WiMAX transmitter. The objective of the latter is to complement the design detailed previously, considering that the MIMO version of the receiver features a scaled version of the baseline adaptive memory architecture. The open-loop version of the MIMO system is only considering the PUSC permutation scheme, whereas the closed-loop features both the PUSC and AMC schemes.

Figure 5.36 presents an outline of the memory-plane designed for the MIMO transmitter, where each block in the diagram includes a specifically modified instance of the extended adaptive memory structure (i.e., memory entity and dedicated controller). For instance, 16-bit I/Q values are utilized throughout the different baseband processing stages of the transmitter (i.e., 1024x16-bit RAM block primitives are used) and dedicated state-machines are utilized by each memory controller for both read and write memory accesses (resulting in the generation of the required global addresses and control signalling). Moreover, an internal register complements the previous by controlling the contents of each memory element comprisng the associated adaptive memory entity. As one should expect, despite these modifications, the core of the adaptive memory structure presented earlier is re-utilized.

Subchannelization stage

The first adaptive memory structure of the multi-antenna transmitter is a scaled-version of the baseline design, which was modified to fit its precise needs. The



(a) Transmit antenna 1. (b) Transmit antenna 2.

Figure 5.37: Allocation of the data symbols for the two transmit antennas according to the matrix A (defined in the WiMAX standard).

memory entity comprises 16 block RAM instances; two 2x1440x16-bit memory elements were used to serve the storage needs of the PUSC permutation scheme. The associated controller implements the required subcarrier-scrambling operations as follows:

(a) The subcarriers are written in an interleaved fashion to the two memory elements, in groups of twenty-four sequentially-ordered modulated symbols (recall the PUSC subchannelization process described by Figure 5.1a).

(b) The subcarrier retrieval is simply performed in sequential order for each memory element (i.e., until it is empty).

Space-time block coding stage

This stage included the creation of the different data streams of each transmit antenna, according to the matrix A STBC configuration described in (5.14). An extended design was therefore employed for the adaptive memory structure utilized in this stage.

The memory entity required twice the storage capacity of the previous stage, since two flows of the same size are generated from the I/Q data; this resulted in four 2x1440x16-bit memory elements. Thus, 32 block RAM instances are required to store a PUSC slot per transmit antenna. Taking advantage of the common low-level address and control signalling design, the increase of memory elements has little impact on the control-plane complexity. Furthermore, as it can be seen in Figure 5.37, the resulting flows for each antenna are composed by crossed and modified versions of the same I/Q samples. More specifically, the second OFDM symbol of each PUSC slot of the second transmit antenna is the conjugated of the first OFDM symbol of the equivalent slot in the first transmit antenna. Similarly, the second OFDM symbol of each slot in the first transmit antenna is the negated and conjugated of the first OFDM symbol of its counterpart slot for the second transmit antenna. Therefore, the internal addressing scheme for the read memory accesses has been modified to apply on-the-fly the OFDM symbol interleaving (i.e., since the whole slot is stored, the reading process crosses the internally selected memory elements to return the I/Q samples required for each antenna). The previous is summarized in Figure 5.38a. The associated controller provides the remaining functionalities:

(a) During the sequential writing of the I/Q samples, the required conjugation and negation of the samples is applied on the-fly, by means of a dedicated circuitry, as seen in Figure 5.38b (i.e., the I/Q samples are written without any modification at one memory element, whereas the modified version is stored at the equivalent memory element utilized for the second data flow).

(b) The data is retrieved in sequential order for each memory element (i.e., recall that the OFDM symbol interleaving is automatically applied in the memory entity).

Permutation and clustering stage

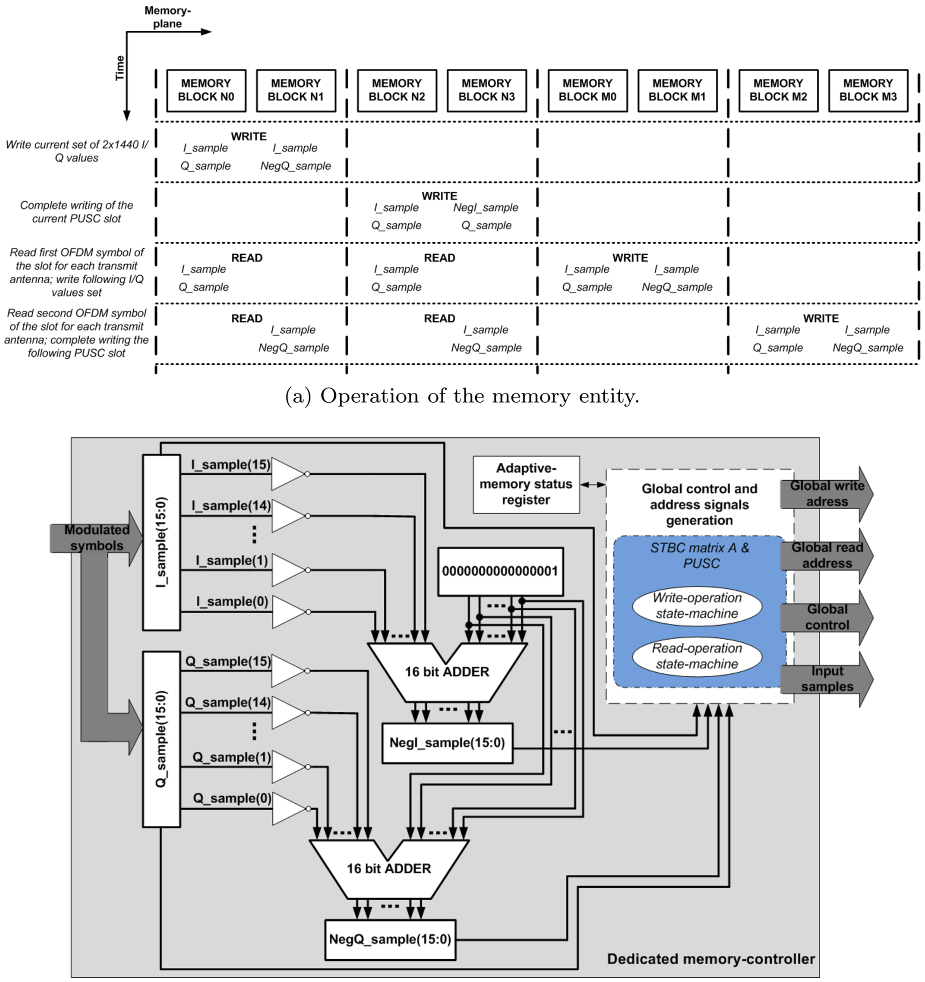
Two design details have to be underlined for the PUSC permutation and clustering operations. First, instead of using a single adaptive memory structure to apply the required scrambling operations for both antennas, it is utilized a simplified design which is replicated for each antenna-processing branch. In other words, the STBC stage constitutes a division point from where N distinct processing branches are utilized (N is the number of transmit antennas). The latter not only results in a simplified dedicated-controller design, but facilitates future extensions of the system (e.g., use more transmit antennas). Moreover, two standard-related operations are implemented in a single storage-stage. To achieve that, the dedicated controller at each antenna-processing branch is based on the following:

(a) The I/Q sample-writing process is following the ordering defined by the PUSC scrambling-formulas (refer to Section 5.3.5 for more details). Thus, the permutation operation is performed during the write access.

(b) Similarly, the generated global read addresses are based on the predefined PUSC clustering-permutation vectors. Hence, the clustering operation is executed during the read access.

Pilot/DC/Guard-Band insertion stage

Prior to the IFFT processing stage, three different standard-related operations are executed. This includes the insertion of the pilot tones jointly with the DC and guard-bands, the mandatory subcarrier randomization (i.e., weighting process) and, finally, a cyclic shift which is applied to the formatted OFDM symbol (i.e., the DC carrier which is originally inserted at position 1024 needs to be situated at position 0 before being processed by the IFFT, accounting for the inherent LO coupling at the RF transmitter). To achieve a 0-latency during the insertion of the pilot subcarriers, it is required to account for the different distributions of every single PUSC slot configuration. Since four possible pilot distribution cases are defined for the 2x2 MIMO PUSC scheme, the equivalent amount of memory elements is utilized (per antenna). Hence, four 1024x16-bit RAM blocks comprise each memory element. The RAM blocks are pre-initialized as shown in Figure 5.39 (i.e., the predefined values of the pilots, DC and guard-band subcarriers are encountered in the positions indicated by the



(a) Operation of the memory entity.

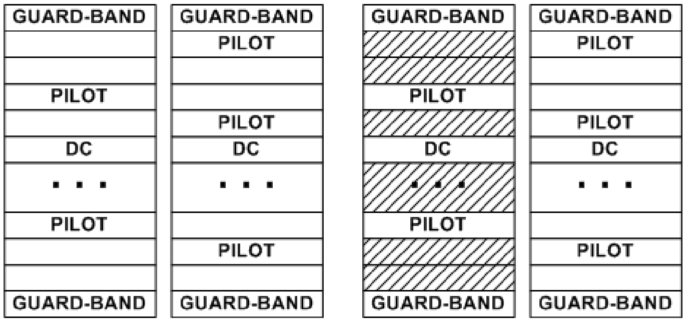
(b) Detail of the dedicated memory controller.

Figure 5.38: Design details of the adaptive memory structure utilized in the STBC stage of the 2x2 MIMO transmitter.

PUSC permutation scheme). Then, the adaptive memory controller acts as follows:

(a) The received I/Q data samples are sequentially written, respecting the predefined pilot/DC/guard-band positions. Hence, the global write address generation is creating the required write-holes (i.e., so as not to overwrite the predefined values). This for instance implies that for the PUSC case, 1440 write accesses are performed per OFDM symbol.

(b) On the contrary, during the read process the complete OFDM symbol (i.e., 2048 positions) is sequentially passed to the FFT. Nonetheless, the generated read addresses are starting from position 1024 to perform the required cyclic shift on-the-fly; the first returned I/Q values are the ones composing the second half of the carriers of the OFDM symbol (thus, the



(a) Dedicated RAM-blocks exist for the different considered slot-formats.

(b) The data samples are written respecting the pre-initialization mask.

Figure 5.39: Illustrative operation case of the proposed 0-latency pilot/DC/guard-band insertion processing stage.

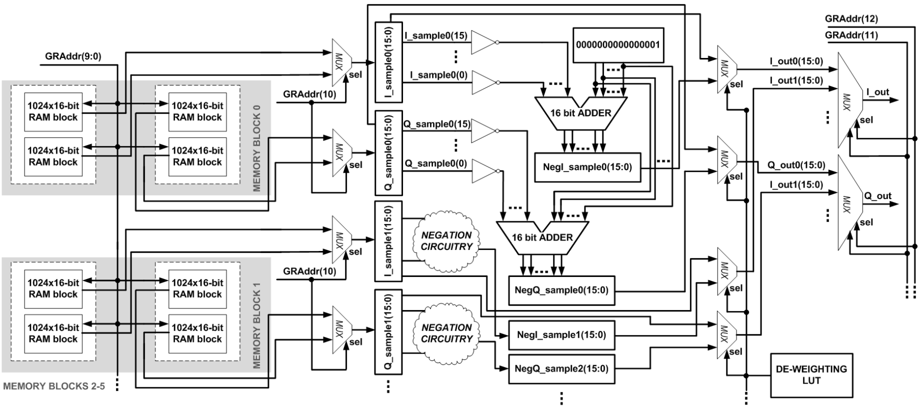


Figure 5.40: Detail of the augmented design of the memory entity to implement the required weighting process.

process ends by reading the first 1024 samples originally written on the block RAM).

On top of that, additional logic was included in the memory entity to perform in a concurrent fashion the weighting process with the read accesses. As it can be observed in Figure 5.40, the PRBS generator utilized to control the randomization process (as defined in the WiMAX standard) has been implemented by means of a pre-initialized LUT. Then, each couple of read I/Q values is negated on-the-fly by dedicated circuitry. Finally, the LUT acts as a mask by driving the MUX that decides whether the original or the negated I/Q values need to be forwarded to the IFFT processing block.

Adaptive memory architecture of the MIMO closed-loop scheme

Another practical manifestation of the incremental development, was the inclusion of the AMC permutation scheme to the 2x2 MIMO configuration of the system. The AMC zones feature a TAS scheme, where groups of adjacent sub-

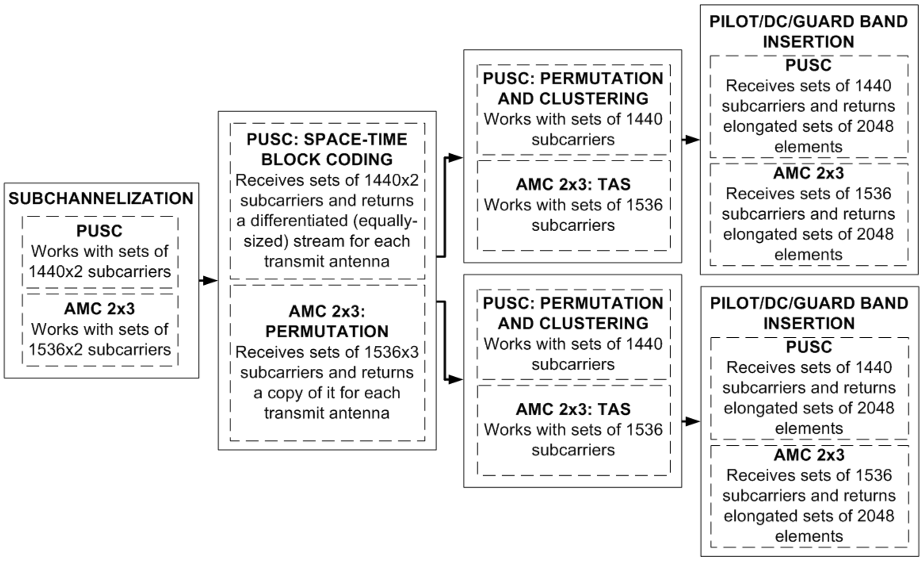

SUBCHANNELIZATION
PUSC
Works with sets of 1440x2 subcarriers
AMC 2x3
Works with sets of 1536x2 subcarriers
PUSC: SPACE-TIME BLOCK CODING
Receives sets of 1440x2 subcarriers and returns a differentiated (equally-sized) stream for each transmit antenna
AMC 2x3: PERMUTATION
Receives sets of 1536x3 subcarriers and returns a copy of it for each transmit antenna
PUSC: PERMUTATION AND CLUSTERING
Works with sets of 1440 subcarriers
AMC 2x3: TAS
Works with sets of 1536 subcarriers
PUSC: PERMUTATION AND CLUSTERING
Works with sets of 1440 subcarriers
AMC 2x3: TAS
Works with sets of 1536 subcarriers
PILOT/DC/GUARD BAND INSERTION
PUSC
Receives sets of 1440 subcarriers and returns elongated sets of 2048 elements
AMC 2x3
Receives sets of 1536 subcarriers and returns elongated sets of 2048 elements
PILOT/DC/GUARD BAND INSERTION
PUSC
Receives sets of 1440 subcarriers and returns elongated sets of 2048 elements
AMC 2x3
Receives sets of 1536 subcarriers and returns elongated sets of 2048 elements

Figure 5.41: Overview of the memory-plane designed for the 2x2 MIMO closed-loop transmitter.

carriers are instantaneously allocated to the antenna that is having the most favourable conditions. Hence, the contents of the transmitted DL frame are dynamically modified. The length of the PUSC and AMC zones is known a priori; the PUSC scheme is basically utilized in the two first OFDM symbols of the frame. This PUSC OFDM symbol pair is transmitted according the STBC-based scheme detailed before and is meant to describe the specific subcarrier allocation utilized in the subsequent AMC zone (i.e., the DL-MAP defined in the WiMAX standard was tailored to fit the needs of the proposed proof-of-concept). The existence of two permutation schemes and two DL transmission modes increases the design and implementation complexity of the adaptive memory structure.

In the transmitter, the adaptive memory entities were scaled to provide the required additional storage capacity. Similarly, the dedicated controllers were extended to account for the AMC-related subchannelization, permutation and weighting processes. Moreover, the standard-related operations are implemented in a different order for the two permutation schemes, in order to optimally exploit the previously defined architecture. A high-level representation of the previous is shown in Figure 5.41. The incremental design features the following:

- A single memory entity is utilized to implement both the PUSC STBC and AMC permutation stages; the dedicated memory controller is in charge of implementing the different operations by adequately generating the global write and read addresses as described earlier in the open-loop transmitter (e.g., two exact copies of the same set of I/Q values is forwarded during the processing of the AMC-formatted OFDM symbols).

- The memory structure and controller are replicated to jointly implement the PUSC permutation (and clustering) with the AMC TAS functionalities. A separated processing branch exists per antenna. For the AMC-formatted OFDM symbols, complementary subsets of the received data-samples are kept at each antenna (i.e., according to the feedback information obtained from the receiver).

Similarly, the de-subchannelization stage at the receiver is a scaled version of the baseline adaptive memory structure of its single-antenna counterpart. In this sense, the storage capacity of the memory entity needs to be doubled (i.e., accounting for the two receive antennas). Besides, the utilization of dual-port memories[5] is required considering the throughput requirements and the configuration of the AMC slot-formation. As detailed in Section 5.3.3, the MRC stage is designed jointly with its STBD counterpart to provide two symbol estimates per clock cycle. On the one hand, the previous results in the utilization of a common access index for the two different block RAMs in the PUSC zone (i.e., recall that in the STBD both $\hat{d}_k[2O_n]$ and $\hat{d}_{k+1}[2O_{n+1}]$ are retrieved together, which - taking into account the subcarrier allocation detailed in Figure 5.37 - results in storing them at the same position in the two different memory elements). Thus, the retrieval of the original symbol sequence requires an interleaved OFDM symbol reading. On the other hand the combination of the applied TAS scheme and the AMC subchannelization process often results in storing consecutive symbols in neighbouring positions within the same memory element. Hence, during the AMC de-subchannelization process, it is frequently required to perform two simultaneous read accesses to a single RAM block.

5.3.5 Implementing operations related to the WiMAX standard without using DSP slices

As detailed in Section 5.1.2 the mobile WiMAX PHY-layer requires various subcarrier grouping and scrambling operations during the formation of the OFDM symbols. This in turn requires the calculation of various indexing values, which often involve non-trivial arithmetic operations such as divisions or modulus-calculations. Considering that the target FPGA devices do not feature a great amount of DSP48 slices (i.e., optimized for arithmetic calculations), it is important to avoid using such slices whenever it is feasible, as in the case of the operations related to the WiMAX standard. Besides, it is a hard performance-requirement to provide a new index each clock cycle so as not to introduce latencies to the processing chain. Taking into account all the previous, it was designed and implemented a fully efficient low-level digital design (i.e., gate-level) of the subcarrier-location calculations (resulting from the utilized permutation schemes). Two indicative examples of this design are included hereafter, illustrating the proposed low-level solutions.

PUSC permutation formula
During the formation or decomposition of the PUSC-formatted OFDM symbols (i.e., permutation and clustering stage of the transmitter and de-subchannelization stage of the receiver) it is required to scramble the incoming modulated symbols. More specifically, to calculate the new position of each I/Q value

[5] Dual-port memories require a larger silicon area, but given the underlying technology of the utilized Virtex-4 FPGA devices, the same amount of embedded memory resources are required for single and dual-port RAMs.

the mobile WiMAX standard defines the following permutation formula for the PUSC-structured OFDM symbols:

$$k(l, c) = n_l \cdot N_{sch} + permbase[(c + n_l) \mod N_{sch}], \tag{5.41}$$

where k is the index of the subcarrier within the OFDM symbol, $c \in \{0, ..., 11\}$ is the subchannel index within the MG, $l \in \{0, ..., 23\}$ is the subcarrier index within the subchannel, $n_l = (l + 13 \cdot c) \mod 24$, N_{subch} is the number of subchannels within the MG (i.e., either eight or twelve, depending on the parity of the MG's index) and *permbase* is a predefined subchannel permutation mask (i.e., defined for both lengths of the MG).

k is the associated subcarrier index that defines the position of the subcarrier in respect to the IFFT/FFT operation, and thus is provided as an input to the logic calculating the formula. The values of l, c and N_{subch} are generated throughout the processing of each OFDM symbol by the internal control plane (i.e., which receives the frame formatting information from the centralized control unit).

In order to calculate the nl term of the formula, it is first required to calculate the $\mod 24$, which in fact represents an equivalent implementation of a division by a constant (i.e., $m \mod 24 = r$, where r is the remainder of the division of m by twenty-four). The design solution takes advantage of the fact that when using fixed-point arithmetic, a division by 2^N can be easily implemented by shifting N positions to the right the decimal point, which eventually yields a hardware-efficient implementation. Indeed, by utilizing the following decomposition, $24 = 8 \cdot 3$ (i.e., $8 = 2^3$), the division by 24 can be expressed as:

$$m = q \cdot 24 + r = (q_1 \cdot 3 + r_1) \cdot 8 + r_0, \tag{5.42}$$

where q_0 and r_0 are respectively the quotient and remainder of the division of m by 8, and $q1$ and $r1$ are respectively the quotient and remainder of the division of q_0 by three (i.e., $m = q_0 \cdot 8 + r_0$ and $q_0 = q1 \cdot 3 + r_1$). Thus, the remainder of the division by 24 can be expressed as:

$$r = r_1 \cdot 8 + r_0 \tag{5.43}$$

The initial division by eight is a low-cost implementation since r_0 is easily obtained by keeping the three LSBs of the fixed-point representation of m, while q_0 is composed by the remaining MSBs. Moreover, the simplest way to implement the division by 3 of the resulting q_0 is by utilizing a LUT. More specifically, taking into account that $m = l + 13 \cdot c$ requires eight bits, q_0 is then comprised by five bits. This results in 32 possible cases for the calculation of r_1, considering that two bits are required for $r1$ and thus it is only needed to store 64 bits, which can be easily realized by four *LUT4* and two *MUXF5* Virtex-4 primitives. Furthermore, only three additional adders are required to provide the final calculation of n_l, as shown in Figure 5.42. As it can be observed, the required multiplication can be also easily implemented by using a number of adders and shift-registers. The remaining calculations of the permutation formula are implemented following the same rationale, resulting in a hardware-efficient implementation that does not need to utilize DSP-slices. The presented gate-level architecture is asynchronous and executed concurrently with the synchronous control-plane generating the required l, c and N_{subch} values (which are registered to ensure the stability and temporal coherence with the generated index values).

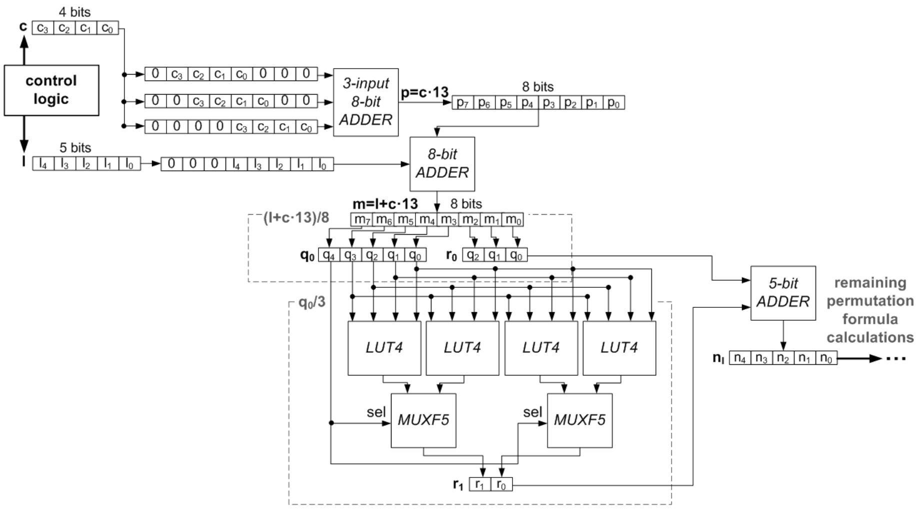


Figure 5.42: Proposed hardware-efficient gate-level design to implement the calculation of n_l.

Data/pilot separation in the AMC slots

The principal objective of the data/pilot separation is to provide two differentiated pilot and data I/Q-flows to the channel estimation processing block. Hence, this processing stage is strongly related to the permutation scheme of each OFDM symbol. More precisely, it is required to identify the exact position of a subcarrier within a given logical structure (i.e., cluster or BIN). Thus, to do so, it is required to decompose the associated FFT output-index, k, in two subindexes: c, identifying the location of the logical structure within the OFDM symbol, and p, providing the location of the subcarrier within the previous.

An illustrative example is detailed hereafter for the calculation of c and p in AMC-formatted slots. Given the length of the defined BINs, the decomposition of k is provided by:

$$k = c \cdot 9 + p, \tag{5.44}$$

where $c \in \{0, ..., 191\}$ and $p \in \{0, ..., 8\}$. Hence, it is required to divide the associated subcarrier-index by nine. A LUT-based design is also considered, but given that k utilizes eleven bits, it would not be the most appropriate to store all possible 2^{11} quotient (i.e., 8-bit c) and remainder (i.e., 4-bit p) values.

An alternative solution is to perform a simplified two-phase calculation, which is based on the bitwise division of k by nine. More specifically, expressing k as the weighted sum of powers of two (i.e., $\sum_{b=0}^{10} k_b \cdot 2^b$), each possible quotient, q_{l_b}, and remainder, r_{l_b}, value resulting from the division by nine of each component of the previous sum (i.e., $k_b \cdot 2^b = q_{l_b} \cdot 9 + r_{l_b}$) can be pre-calculated. The obtained values are shown in Table 5.4.

Then, the intermediate quotient and remainder values of the division of k by nine, obtained in the first phase, can be defined as:

$$q_{first} = \sum_{b=0}^{10} k_b \cdot q_{l_b}, \tag{5.45}$$

b	2^b	q_{l_b}	q_{l_b}
0	1	0	1
1	2	0	2
2	4	0	4
3	8	0	8
4	16	1	7
5	32	3	5
6	64	7	1
7	128	14	2
8	256	28	4
9	512	56	8
10	1024	113	7

Table 5.4: Quotient and remainders of the bitwise division of k by nine.

$$r_{first} = \sum_{b=0}^{10} k_b \cdot r_{l_b} \tag{5.46}$$

Not surprisingly, r_{first} can result in a value larger than nine. Thus, in the second phase it is required to divide it by nine, to refine the previously calculated quotient and remainder values. This second division can be implemented utilizing a similar approach to the previously defined modulus-calculation (given the lower number of possible division cases). Specifically, the division by nine calculated in the second phase can be expressed as follows:

$$r_{first} = q_{second} \cdot 9 + r_{second} \tag{5.47}$$

Finally, the correct quotient and remainder values can be obtained as indicated in the following equations:

$$c = q_{first} + q_{second}, \tag{5.48}$$

$$p = r_{second} \tag{5.49}$$

The calculation of (5.45) and (5.46) can be simplified by considering the values provided in Table 5.4 (i.e., bitwise products by known constant values). The simplified binary-formulations are the following:

$$q_{first} = k_{10} \cdot 1110001_2 + k_{9:6} \cdot 111_2 + k_5 \cdot 11_2 + k_4, \tag{5.50}$$

$$r_{first} = k_{10} \cdot 111_2 + k_{9:6} + k_5 \cdot 101_2 + k_{3:0}. \tag{5.51}$$

As in the previous case, a series of shift-and-add operations are sufficient to calculate both (5.50) and (5.51). Then, a pre-initialized LUT is providing the division by nine of r_{first}. In more detail, accounting for the worst possible case, r_{first} requires six bits, since the number of quotient and remainder values required to be stored is reduced to 64. Figure 5.43 depicts the FPGA architecture utilized to obtain the c and p terms of the AMC-formatted OFDM symbols (avoiding the use of DSP48 slices).

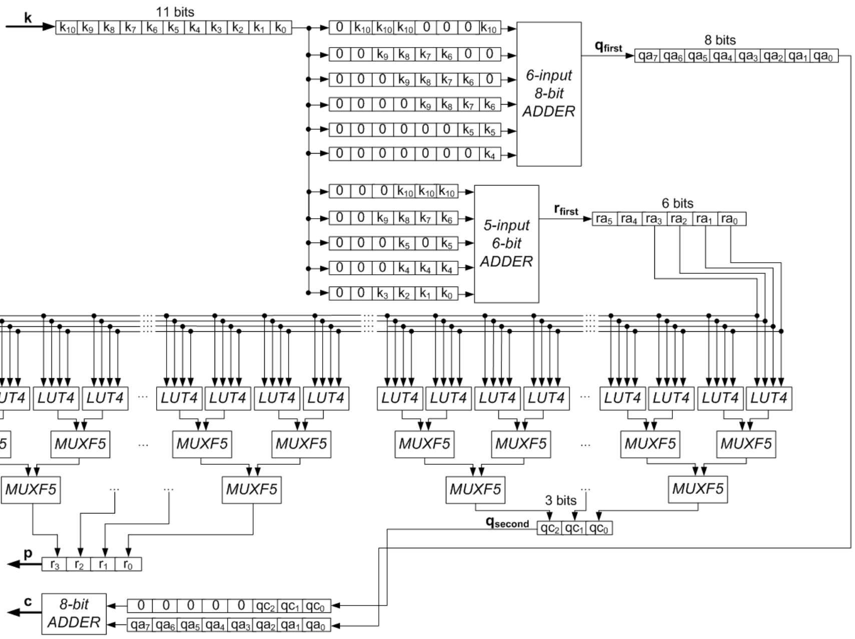


Figure 5.43: Low-level design to implement a division by 9.

5.3.6 Centralized control unit

The efficient PHY-layer implementation of a communication system that features both a MIMO open-loop configuration and a TAS-based closed-loop scheme requires a processing architecture able to apply FPGA-resource sharing in a great extent. Various other signal processing stages, such as the FFT, need to be utilized by different processes that run both schemes in different time instants. Therefore it is essential to provide a centralized control unit to serve the operation of the different processing elements comprising the PHY-layer and control its synchronous communication in a strict timing manner.

This section is focuses on the control unit encountered in the MIMO closed-loop transmitter (i.e., the receiver of this system features a simplified version of the transmitter structure). Two main aspects define the complexity of the control plane; i) the configurable parameters related to the mobile WiMAX standard operations and ii) the dynamic subcarrier allocation due to the utilized TAS scheme.

As indicated in Section 5.2, various DL frame-parameters are fixed (recall Table 5.2) and, thus, the central control unit generates the required control signals utilizing a set of predefined values. Nevertheless, it has to be stressed-out that the designed PHY-layer architecture is prepared to allow the dynamic configuration of those parameters, since a dedicated port in various processing blocks allows to program their internal logic, which was designed to work with more than one operation-mode (i.e., fixed and adaptive). A detailed list of the PHY-layer parameters controlled by this central unit follows:

- *CP length:* the specified frame format utilizes a fixed CP length equal to one quarter the OFDM symbol size. Thus a single value is provided to the control signals responsible for configuring the CP insertion stage. Nonetheless, the related logic supports all CP lengths defined in the standard.

- *IFFT size:* another flexible PHY-layer parameter of the WiMAX standard is the baseband signal bandwidth which was fixed at 20 MHz. This consequently resulted in a fixed IFFT size of 2048 points. However, the Xilinx IFFT core can be dynamically reconfigured to provide any of the bandwidth sizes defined by the WiMAX standard. The central control unit generates as well the additional signalling that manages the operation of the IFFT core.

- *WiMAX standard-related operations:* as indicated earlier, the precise functions to be executed in each processing stage related to operation defined in the WiMAX standard, depends on the format of the WiMAX frame. Hence, the central control is in charge of the timely configuration of those processing stages that need to be adapted according to the dynamically modified frame-format. As in the previous cases, the predefined frame formats allow to know a priori the exact number of PUSC and AMC OFDM symbols to be generated (i.e., the precise control-values at each time instant are predefined as well).

It is important to underline that only a moderated extension of the control plane is required to achieve a fully-dynamic and flexible PHY-layer configuration (e.g., extension of the custom DL-MAP structure, replacement of the current padding-bit zone with additional control-information to allow the adaptation of the mentioned parameters). By all means, the RTL design of the central control unit provides a solid and modular basis that allows to expand the current closed-loop system by adding more PHY-layer parameters or configurations. It also consists the core component that enables the FPGA resource sharing design strategy.

In the TAS scheme the subcarrier allocation is dynamically adapted according to the CSI provided by the receiver. More specifically, the TAS decision is performed at the receiver and communicated to the transmitter by means of a dedicated feedback link (recall Section 5.2.4). The feedback information plays a crucial role in the control plane of the closed-loop transmitter. While this feedback is received in a non-determined time instant at the beginning of the inter-frame silence periods, the generation of the frames to be transmitted requires the timely-execution of the different baseband DSP-stages. In other words, the operation of each processing block at the transmitter must comply with a strict time budget. This means that the control unit accounts for the internal latency of each separate processing block, and also of the entire baseband as a whole, to create an uninterrupted flow of OFDM symbols which is eventually interfaced with the input of the DAC circuitry. The received feedback information is rapidly processed and stored in dedicated registers, which are later accessed to properly configure the adaptive PHY-layer operation. Furthermore, these registers are initialized with predefined values, allowing the operation of the transmitter when no feedback information is received; the latter results in a subcarrier allocation based on the STBC-based open-loop configuration. Specifically, the received feedback is utilized to command the operation of the following DSP stages:

- *DL-MAP generation:* in order to generate the OFDM-symbols comprising the DL-MAP, the central control unit forwards in a timely manner the feedback to the processing block in charge of generating these two PUSC OFDM symbols (i.e., the custom DL-MAP structure is basically an extension of the received feedback, as seen in Figure 5.20). The control unit is also managing the multiplexing of the resulting DL-MAP bit-sequence with the one produced by the PRBS generator (i.e., user-data).

- *PRBS generator:* first it is required to generate the padding-bits of the last PUSC OFDM symbol of the DL-MAP. Thus, the central control forces the generation of 1200x2 bits (i.e., recall that in the PUSC zone the QPSK modulation is mandatory). Then, the pseudo-random bits of the user-data need to be generated accounting for the possibility that different symbol mappings can be utilized (i.e., per 16-subcarrier set). The control plane then parses the feedback information to indicate the length of the different bit-sequences required by each posterior modulated symbol.

- *Configurable mapping:* the same information utilized to configure the PRBS generator is also utilized for the configurable mapping stage[6] (i.e., the QPSK modulation is applied to the PUSC bit sequence, while the user-requested modulation is applied to the remaining bits).

Figure 5.44 provides a detailed graphical summary of the central control unit design. As it can be observed, dedicated state machines - relying on predefined threshold-values and PHY-layer parameters - form the backbone of the implemented control-architecture. Taking into account that the principal goal of the baseline design (i.e., initial design iteration) was to provide a robust and latency-optimized control plane, a simple internal communication protocol was designed; the latter only requires to indicate when valid data enters each processing block or stage (the same logic acts as a trigger for the operation of each processing block). Moreover, the internal delays of the inter-block communications are accounted during the generation of the control signals and data-forwarding (i.e., whenever it is possible those processes are executed N clock cycles before the resulting inputs are required by the corresponding processing stage, where N is the pre-calculated intra-block communication delay). Likewise, they are avoided unnecessary additional latencies.

5.4 Integration and implementation using the GEDOMIS testbed

The produced custom HDL code was simulated utilizing the Mentor Graphics ModelSim 6.3a RTL simulator. The latter is completely integrated with the Xilinx ISE 9.2i design suite which was utilized to produce the FPGA implementations. Furthermore, the latter has facilitated the utilization of Xilinx IP cores and the insertion of FPGA-monitoring blocks (i.e., ChipScope Pro).

[6] The implemented configurable modulation can be considered the first step towards a fully-adaptive modulation scheme. Moreover, the designed architecture can readily implement such functionalities (e.g., by including the calculation of an additional SNR-related metric in the receiver).

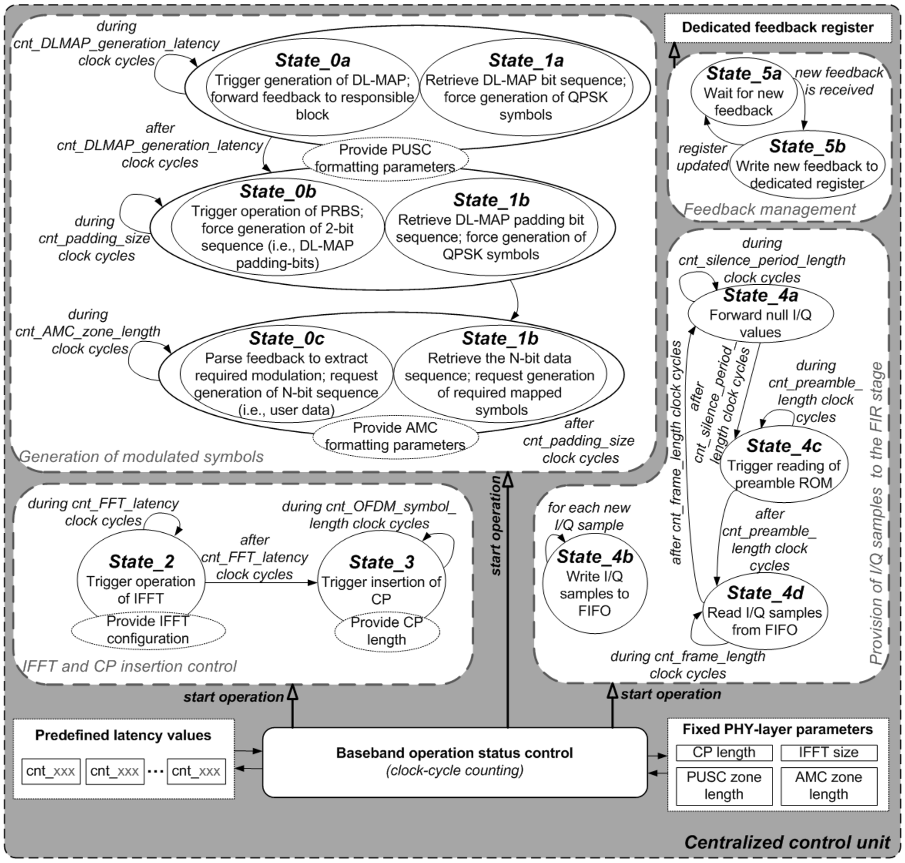

during
cnt_DLMAP_generation_latency
clock cycles
State_0a
Trigger generation of DL-MAP; forward feedback to responsible block
State_1a
Retrieve DL-MAP bit sequence; force generation of QPSK symbols
after
cnt_DLMAP_generation_latency
clock cycles
Provide PUSC formatting parameters
during
cnt_padding_size
clock cycles
State_0b
Trigger operation of PRBS; force generation of 2-bit sequence (i.e., DL-MAP padding-bits)
State_1b
Retrieve DL-MAP padding bit sequence; force generation of QPSK symbols
during
cnt_AMC_zone_length
clock cycles
State_0c
Parse feedback to extract required modulation; request generation of N-bit sequence (i.e., user data)
State_1b
Retrieve the N-bit data sequence; request generation of required mapped symbols
Provide AMC formatting parameters
after
cnt_padding_size
clock cycles
Generation of modulated symbols
Dedicated feedback register
State_5a
Wait for new feedback
new feedback is received
register updated
State_5b
Write new feedback to dedicated register
Feedback management
during
cnt_silence_period_length
clock cycles
State_4a
Forward null I/Q values
after cnt_frame_length clock cycles
after
cnt_silence_period
length clock cycles
during
cnt_preamble_
length clock
cycles
State_4c
Trigger reading of preamble ROM
after
cnt_preamble_
length clock
cycles
State_4d
Read I/Q samples from FIFO
during cnt_frame_length
clock cycles
Provision of I/Q samples to the FIR stage
for each new
I/Q sample
State_4b
Write I/Q samples to FIFO
start operation
during cnt_FFT_latency
clock cycles
State_2
Trigger operation of IFFT
Provide IFFT configuration
after
cnt_FFT_latency
clock cycles
during cnt_OFDM_symbol_
length clock cycles
State_3
Trigger insertion of CP
Provide CP length
IFFT and CP insertion control
start operation
start operation
Predefined latency values
cnt_xxx
cnt_xxx
cnt_xxx
Baseband operation status control
(clock-cycle counting)
Fixed PHY-layer parameters
CP length
IFFT size
PUSC zone length
AMC zone length
Centralized control unit

Figure 5.44: Detailed representation of the designed centralized control unit.

This section aims at describing the integration of the baseband design to the firmware of the ADP boards utilized by GEDOMIS (recall Section 4.4), which is required before the final FPGA-programming bitstreams can be produced. Additionally, it is also detailed the precise testbed setup utilized to deploy the presented systems.

5.4.1 Real-time MIMO signal transmission

Once the baseband RTL design of the MIMO-OFDM mobile WiMAX transmitter was verified (VHDL versus Matlab co-simulations), a detailed study was conducted to provide its integration with the DAC circuitry of the target boards. The text below is particularized for the two-antenna transmission schemes.

First, after the insertion of the CP, the generated OFDM symbols (now consisting of 2560 samples) must pass through latency-absorption FIFO memories in order to provide an uninterrupted flow of data to the DAC devices, accounting for the variation of the length of the subcarrier sets during the operation of the baseband (e.g., different slot definitions, insertion of the pilot tones, DC, guard-bands or CP), which originated latency-gaps between the OFDM symbols comprising the frame to be transmitted. Additionally to the FIFO, in the first transmit antenna it was also utilized a dedicated ROM, which stored the preamble (i.e., predefined I/Q values). The ROM contents were read just before the first FIFO outputs were available.

In order to produce the global synthesized IF signal and take the maximum advantage of the filters contained in the DAC devices of the VHS-DAC board, an appropriate filter-interpolation strategy is required. The previous is a significant design, implementation and system-integration decision, which in fact has to be carefully considered every time a designer of MIMO-OFDM transmitters intends to interface the baseband design with a DAC device. The filter interpolation is an excessively processing-consuming implementation for FPGA devices, having a very hard to define trade-off between of FPGA resources utilization (i.e., embedded memory blocks and DSP slices versus generic FPGA logic) and maximum achievable clock rate. The heavily populated FPGA designs, with multiple clock domains, are making the PAR process of the FPGA implementation-flow a particularly hard task. Therefore, a key system-design objective is to migrate the challenging implementation of the interpolating filters from the FPGA domain to high-performance versatile DAC devices.

The VHS-DAC board is equipped with dual DAC devices, which are using 3 filters capable of applying a x8 interpolation. Considering that, to produce an IF signal centered at 67.2 MHz, a total x16 interpolation is required for each of the two I/Q data-streams of the transmitter, it was selected to apply a x2 interpolation in the FPGA domain and a x8 interpolation in each dual DAC device, which are clocked at 44.8 MHz.

In more detail, the I/Q 16-bit data at the output of the latency-absorption FIFOs are fed to four separate filters (hosted in the FPGA device), which apply an interpolation by two. The filters are clocked at 44.8 MHz (i.e., the baseband clock is at 22.4 MHz). The use of two separate filters for the I/Q data components of each antenna was made to avoid the multiplexing of data, which requires a double clock rate from the one currently used at baseband (i.e. increasing the input clock from 44.8 MHz to 89.6 MHz). This decision was made after realizing that, what was gained in terms of FPGA-slice area (by using two instead of four interpolation filters clocked at 89.6 MHz), was resulting in a complicated PAR process, which is prone to timing errors (i.e., the FPGA routing algorithms are stressed to the limits for achieving the desirable clock rate). Manual placing of the implemented logic to the FPGA device area is another design option that is equally complicated when considering a large FPGA design with numerous digital processing components.

MATLAB was used to design the filters with a high-rejection response of 80 dB, as depicted in Figure 5.45. While this implies the utilization of 76 real filter coefficients (i.e., high computational cost), its symmetric response helps minimizing the required DSP-blocks within the FPGA. Finally, the FIR IP core from Xilinx was utilized to implement the required filters. Two configuration-aspects deserve of further detail. Conveniently for the system designer, the coefficients file generated by MATLAB can be utilized directly in the Xilinx Core Generator tool. Moreover, accounting for the memory-demanding architecture of the previous transmitter baseband stages it was selected a distributed arithmetic architecture (i.e., while it is not fully taking advantage of the capacities of the available DSP-blocks, it minimizes the embedded memory-block utilization). Finally, the resulting 28-bit output of the I and Q filters is appropriately truncated to allow the utilization of the full dynamic range of the baseband part of the

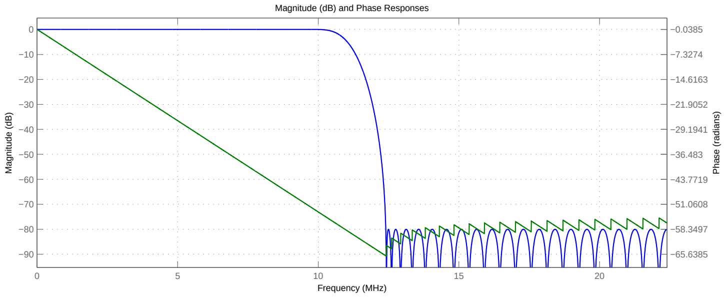


Figure 5.45: The magnitude and phase response of the x2-interpolation pre-DAC interpolation filters used in the transmitter.

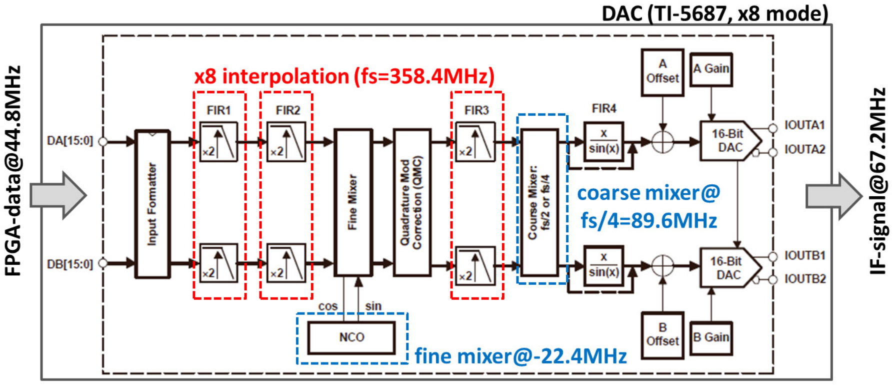
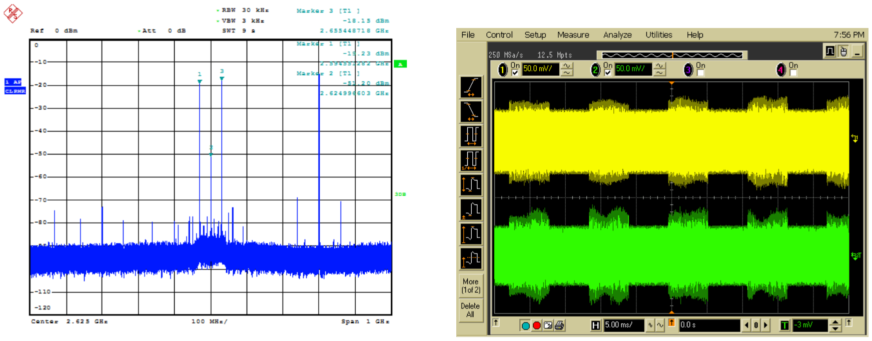


Figure 5.46: Internal configuration of the DAC 5687 chips to attain a x8 interpolation.

transmitter (i.e., further co-simulations were conducted to obtain its optimum fixed-point representation).

A custom C-based application was developed to allow the read/write access of several internal registers of the DAC 5687 chips on-the-fly (i.e., the x8 FMIX QMC CMIX mode is used [DAC]). Specifically, a x8 interpolation results in a DAC sampling frequency, f_s, of 358.4 MHz, considering that the ADC stage is clocked at 44.8 MHz. Taking into account the pass-bands of the different interpolation filters within the DAC, the frequency response of the inverse-sinc filter, and the frequency bands of the signals and their aliases, the coarse mixer was set to 89.6 MHz (equal to $\frac{f_s}{4}$, where $f_s = 358.4$ MHz) and the fine mixer was set to -22.4 MHz. The registers corresponding to the Numerically Controlled Oscillator (NCO) frequency, the fine mixer gain and the coarse mixer mode-configuration are programmed accordingly (i.e., cm_mode(3:0) = 1000). The overall DAC configuration can be observed in Figure 5.46.

Finally, the IF outputs generated by the DACs were fed to two Agilent ESG4438C VSGs, which up-converted the signals and provided the RF outputs centred at 2.595 GHz.



(a) Spurious response.

(b) Noisy RF output.

Figure 5.47: RF signal impairments introduced by the channel emulator.

5.4.2 Real-time MIMO channel emulation

The EB Propsim C8 channel emulator was configured to provide mobile MIMO channels based on both the ITU Vehicular A (up to 120 km/h) and Pestrian B (up to 3 km/h) models, for which a predefined channel impulse response configuration is included in the internal software libraries.

A relevant aspect requiring further details is that, when configuring the channel emulator, the selected center RF frequency does not match the one defined for the transmitter. Specifically, a ±35 MHz shift with respect to the transmitter's RF center frequency (i.e., the channel emulator features a 70 MHz bandwidth) is required to avoid DC-offset and I/Q modulation imperfections. Consequently, the center frequency of the RF output of the channel emulator has been set to 2620 MHz. In Figure 5.47a it can be observed the spurious response of the channel emulator for an RF signal centred at 2595 MHz, when utilizing the described shifted RF output configuration. Additionally, in Figure 5.47b it can be seen that there is a very high noise level at the output of the channel emulator, caused by coupling with the LO signal. This coupling will be eventually filtered by the RF downconverters, as detailed below.

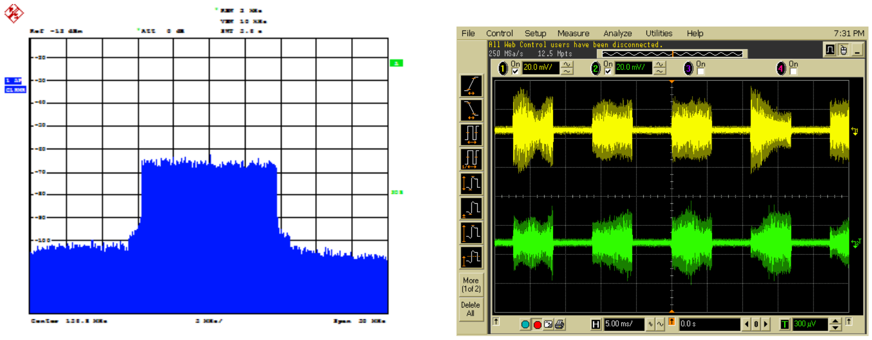
Finally, to enable the empirical validation of our MIMO configuration, it were also utilized two AI NS-3 RF Noise Sources. The delivered noise power levels were accordingly calibrated and balanced for every 2 dB attenuation step.

5.4.3 Real-time MIMO signal reception

The tasks performed at the receiver's RF front-end are the low-noise amplification, the downconversion from RF to IF and the suppression of out-of-band unwanted signals such as noise, spurs and interferences. The power-level of the IF signal needs to be adjusted by the AGC to take full advantage of the system's dynamic range, before being digitized (i.e., ADC) and translated to baseband (i.e., using a DDC as previously detailed). The specific configuration utilized to implement the detailed functions is described below.

RF front-end and ADC

The downconversion stage was provided by the MCS Echotek Series RF 3000 Tuners, resulting in an IF signal centred at 156.8 MHz, for which the LO cou-



All Web Control users have been disconnected.
250 MSa/s 12.5 Mpts
1 On ☑ 20.0 mV/
2 On ☑ 20.0 mV/
3 On □
4 On □
H 5.00 ms/
0.0 s
T 300 µV

(a) RF signal spectrum.

(b) Time-domain MIMO signals.

Figure 5.48: Filtered RF signal at the output of the downconversion stage.

pling resulting from the utilization of the channel emulator has been filtered out. Additionally, considering that the IF-bandwidth of the tuners is higher than the chosen signal bandwidth (i.e., 65MHz versus 20MHz), a prototyped SAW filter was added to optimally match the signal bandwidth at IF and, thus, reduce the out-of-band noise and interferences, likewise eliminating various spurious effects introduced by the channel emulator. In Figure 5.48 it can be observed the filtered RF signal, when an ITU Vehicular channel model at 60 km/h is applied.

Each active component in the receiver chain has a limited dynamic range. Thus, signals exceeding this range are subject to saturation or clipping. Since saturation is a detrimental factor of the system-performance, countermeasures should be taken to prevent it. Moreover, the active RF or baseband processing components are subject to thermal or other types of noise. On top of this, the dynamic range in the baseband part of the receiver may also be potentially affected by the presence of a DC offset (e.g., static DC offsets occur due to bias mismatches in the baseband boards).

Sampling an analog signal at IF results in replicas of the signal's spectrum repeated at uniform intervals (equal to the sampling frequency). The choice of the sampling rate of such signals is dependent on the signal's bandwidth and the IF center frequency. The chosen bandpass sampling architecture required only one ADC for the final IF to baseband conversion.

The ADC uses IF under-sampling and a sampling rate of f_s = 89.6 MHz. Therefore, after the ADC, one of the aliases of the discrete signal will be located at 22.4 MHz, which is the sampling frequency at the baseband receiver, as defined in the WiMAX standard. The digital spectrum after such sampling is depicted in Figure 5.49. The delta at baseband (and integer multiples of the sampling frequency) represents the DC coupling of the baseband hardware which was taken into account in the design of the baseband signal processing stages. The delta at the center of the signal spectrum represents the coupling of the analog LOs at both the up and down-converters.

AGC

The AGC is an analog-digital hybrid processing block providing an interface between the FPGAs and the RF front-end. The PGA, in charge of modifying

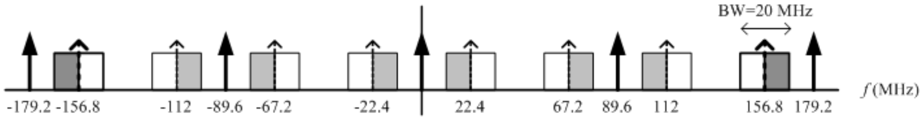


Figure 5.49: Spectrum of the digital signal after sampling at 89.6 MHz.

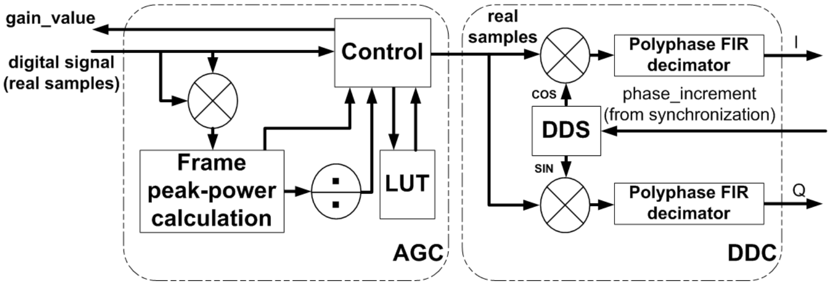


Figure 5.50: Implemented AGC and DDC stages.

the power of the received IF signal, is a digitally-controlled analog circuit with a discrete set of possible gain values, while the algorithm that decides the new gain value of the PGA is implemented in the digital domain.

The correct operation of the AGC is a decisive factor for the overall performance of a mobile receiver. The AGC adjusts in a timely manner the power-level of the input IF signal to utilize the full dynamic range of the ADC and overcome the variations caused by the mobile channel fading. Frame-based OFDM systems are specially prone to high PAPR; it is therefore a prerequisite the inclusion of back-off margin that prevents signal clipping. The ADC device is indicating its saturation with a state signal. When saturation occurs the AGC does not forward data until the signal is attenuated at an optimal dynamic range.

To heart of the AGC algorithm is a signal peak-detector, operating in a per-frame basis (i.e., fixed gain for an entire frame), which provides a baseline trade-off between implementation complexity and efficiency. In a frame-based communication system like mobile WiMAX, where the channel varies rapidly in high mobility conditions, the AGC algorithm has a very limited timing budget to operate. This is because the AGC must calculate the gain of the next frame and apply it to the PGA registers during the inter-frame silence period (i.e., taking into account the peak value of the previous data frame).

The LT 5514 PGA used in our receiver has 16 gain steps with 1.5 dB of separation (i.e., resolution of the gain corrections). Thus, starting from the gain value applied during the previous frame, G_{prev}, the optimal adjustment of the IF input power-level during the following frame, ΔG, in dB is calculated as follows:

$$\Delta G = 10 \cdot \log_{10} g = 10 \cdot \log_{10} \frac{(\frac{v_{FS}^2}{v_{BM}^2})}{v_{PK}^2}, \tag{5.52}$$

where v_{FS} accounts for the digital full scale of the quantizer in the ADC (e.g., 14 bits in our case, using the Q2.14 format), v_{BM} accounts for the back-off safety margin (e.g., 1.3 is the maximum allowed v_{BM} value in our case), and $v_{PK}^2 = \max\ |c_i[n]|^2$, with $c_i[n]$ representing the samples at the output of the

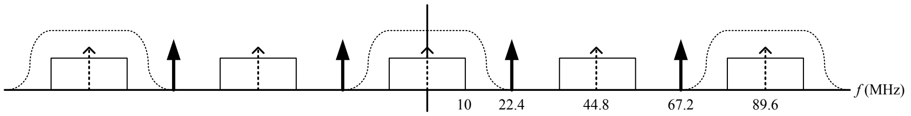


Figure 5.51: Frequency representation of the operations performed in the DDC.

ADC during the previous frame. The precise PGA gain-step to be applied to the IF signal during the following frame, G, can be then expressed as follows:

$$G = \begin{cases} G_{max}, & if \;\; G_{prev} + \Delta G > G_{max} \\ G_{prev} + n \cdot GS, & if \;\; G_{min} \leq G_{prev} + \Delta G \leq G_{max} \\ G_{min}, & if \;\; G_{prev} + \Delta G < G_{min} \end{cases} \tag{5.53}$$

where G_{min} (dB) and G_{max} (dB) represent the minimum and maximum gains of the amplifier.

In order to minimize the processing complexity and the implementation latency it was utilized a pre-calculated LUT (Figure 5.50). More specifically, the LUT represents the threshold values of g in relation to the applicable gain-corrections (i.e., number of gain-steps to be increased or lowered, ΔG).

DDC and CFO correction

Besides the previously detailed channel frequency translation (recall Section 5.2), the DDC implements two additional functions: I/Q components extraction and signal decimation. A high-rejection low-pass FIR filter is responsible for eliminating out-of-band components and isolate the I/Q components (Figure 5.50). Furthermore, an output decimator and formatter selects one out of every four samples, delivering the complex representation of the digitalized signal. Figure 5.51 shows the frequency domain representation of the operations performed within the DDC.

The digital filtering stage of the DDC is important because it prevents aliasing during the sub-sampling process. Hence, it is critical to account for the system-wide signal impairments when designing the digital filter. For this reason, the bandpass and reject frequencies should be carefully selected keeping the useful signal spectrum intact, while at the same time eliminating the effects of the DC level which is an inherent feature of the baseband processing boards (e.g., the DC is transformed to a synchronization-altering sinusoid after being processed by the DDC) and the non-desired alias. After exhaustive MATLAB simulations based on experimental data (i.e., obtained using an offline prototyping approach), it was designed a low-pass filter with 103 coefficients (as shown in Figure 5.52), which was implemented jointly with the decimation stage as a polyphase decimator filter.

Regarding the CFO correction, on power-up the f_{DDS} was tuned at 22.4 MHz and then it was constantly updated in real-time to compensate the effects of the CFO, according to (5.3). In the utilized Xilinx DDS IP core, the output frequency is controlled by a phase increment, $\Delta\theta$, as described in the equation that follows:

$$f_{DDS} = \frac{f_s \cdot \Delta\theta}{2^{B_{\theta(n)}}} \; Hz, \tag{5.54}$$

where f_s is the ADC sampling rate (i.e., 89.6 MHz) and $B_{\theta(n)}$ represents the

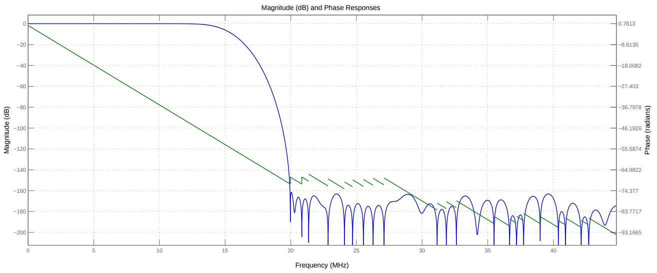


Figure 5.52: The magnitude and phase response of the digital filtering stage of the DDC.

resolution in bits of the internal accumulator used in the DDS (i.e., as part of the co-simulations with the RTL code, this value has been fixed at 32). Then, by combining combining (5.54), (5.3) and (5.4) the phase increment is given by:

$$\Delta\theta = (22.4 + \alpha \cdot \frac{22.4}{2048}) \cdot \frac{2^{32}}{89.6} = 2^{30} + \alpha \cdot 2^{19} \qquad (5.55)$$

Not only the implementation of (5.55) was straightforward, but the resulting value was directly provided to the utilized Xilinx IP core to dynamically update f_{DDS}.

5.4.4 FPGA-based implementation of the presented systems

Accounting for the prototyping capacities of the GEDOMIS testbed (i.e., provided by the ADP boards) all open-loop system configurations required three Virtex-4 FPGA devices to be implemented. On the contrary, the additional complexity of the closed-loop PHY-layer scheme resulted in a five-FPGA realization, as shown in Figure 5.53. As it can be observed, the feedback channel was implemented through a dedicated inter-FPGA communication bus (i.e., emulated UL). Moreover, although it is not included in the figure, in order to select the desired symbol modulation scheme (recall that in the current configuration it can be programmed in a per-frame basis), it was utilized a user-controlled register, which can be modified at run-time through the VHS-ADC's GUI and that is directly mapped to the FPGA device implementing the receiver's feedback generation block.

The precise FPGA-resource utilization of each system configuration is provided in Table 5.5.

Whereas the FPGA-based design of the mobile WiMAX PHY-layer did not follow an energy-efficiency design-path, it might result useful to include preliminary power-consumption metrics to enable a relative assessment of our prototype's power-consumption footprint. It is important to take into account that the target FPGA devices do not belong to a power-efficient family of Xilinx, while at the same time the power-reduction margin that could be achieved by the Xilinx ISE 9.2 tool is very limited.

Indicatively, the Xpower software tool of Xilinx was used to estimate the power consumption of the designed open-loop MIMO mobile WiMAX receiver.

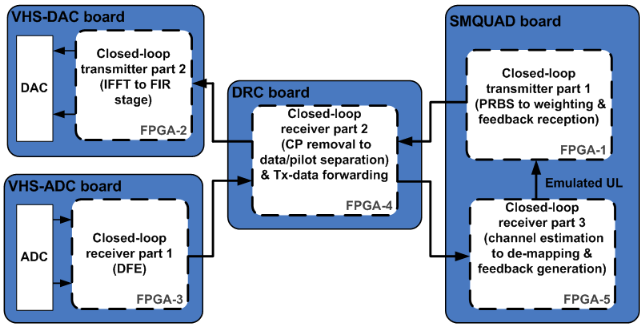

VHS-DAC board
DAC
Closed-loop transmitter part 2 (IFFT to FIR stage)
FPGA-2
VHS-ADC board
ADC
Closed-loop receiver part 1 (DFE)
FPGA-3
DRC board
Closed-loop receiver part 2 (CP removal to data/pilot separation) & Tx-data forwarding
FPGA-4
SMQUAD board
Closed-loop transmitter part 1 (PRBS to weighting & feedback reception)
FPGA-1
Emulated UL
Closed-loop receiver part 3 (channel estimation to de-mapping & feedback generation)
FPGA-5

Figure 5.53: Multi-FPGA partitioning of the implemented closed-loop system.

	Transmitter		Receiver		
	FPGA 1	FPGA 2	FPGA 3	FPGA 4	FPGA 5
	XC4VLX160	*XC4VLX160*	*XC4VLX160*	*XC4VSX35*	*XC4VLX160*
SISO open-loop system configuration					
Slices	-	38%	36%	-	33%
DSP48s	-	43%	37%	-	97%
RAMB16s	-	71%	25%	-	28%
1x2 SIMO open-loop system configuration					
Slices	-	38%	53%	-	52%
DSP48s	-	43%	57%	-	100%
RAMB16s	-	71%	72%	-	51%
2x2 MIMO open-loop system configuration					
Slices	-	77%	53%	-	81%
DSP48s	-	84%	57%	-	100%
RAMB16s	-	98%	72%	-	92%
2x2 MIMO closed-loop system configuration					
Slices	30%	68%	53%	68%	77%
DSP48s	2%	82%	57%	41%	100%
RAMB16s	86%	67%	33%	31%	98%

Table 5.5: Utilization indicators of the FPGA implementation.

The conducted analysis accounted for both quiescent and dynamic power dissipation[7]. Obviously, the design decision to divide the PHY-layer implementation of the mobile WiMAX receiver in two FPGA devices, is playing a major role in the presented power-consumption metrics. The first part, namely the DFE (from the AGC up to the synchronization processing block) is always operating, while the second part (from the CP removal up to the de-mapping processing block) is not operating during the silence period between data frames, which can be considered as the system's idle state. As it may be observed in Table 5.6, when the system is in idle state, the second part of the design (denoted as FPGA-5) will only present quiescent power consumption, achieving a 14% of power consumption savings. The presence of multiple clock regions in FPGA-3

[7] The power consumed when no signal-switching occurs is defined as quiescent, while the dynamic power consumption represents the accumulated power dissipation of the operating components comprising the FPGA design.

Consumption	Quiescent	Dynamic	Total
FPGA-3	1.22 W	1.31 W	2.53 W
FPGA-5	1.13 W	0.62 W	1.75 W

Table 5.6: FPGA-deployed MIMO open-loop receiver baseband power consumption estimation.

increases the overall power consumption. As expected, the power consumption in idle state is quite high due to the inherent high power-leakage of the Virtex-4 FPGA-technology.

5.5 Experimental results

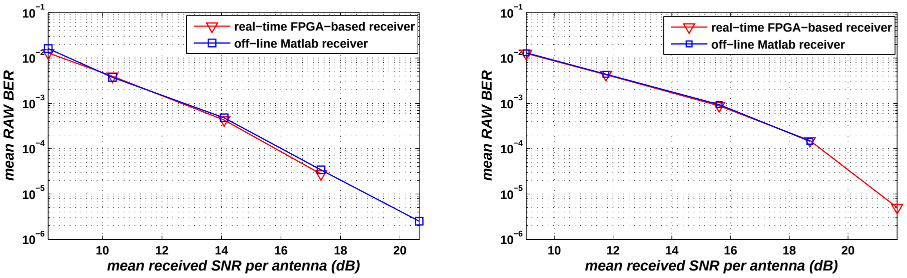
To verify the implemented prototypes, the multi-stage testing stragegy presented in Section 4.2.1 was followed, providing a thorough characterization of the proposed designs at all required levels. Considering that the most complicated scenario was utilizing a 2x2 MIMO antenna-configuration, up to four uncorrelated multipath fading channels were generated (i.e. created with different distribution seeds) based on the ITU Vehicular A or the ITU Pedestrian B channel models [(ITU), 1997]. Furthermore, the channel emulator required to be tuned to provide optimal performance in terms of noise floor, dynamic range and Error Vector Magnitude (EVM), allowing sufficient safety margin for its operation. This was necessary to avoid signal-distortion, degradation of the received SNR, saturation of its ADC and DAC stages (e.g. the transmitted signal PAPR has to be accounted) and quantization errors.

Starting with the open-loop systems, a limited performance-comparison between the resulting real-time FPGA-based prototypes and its off-line MATLAB-based counterparts allowed to assess the implementation losses (e.g., quantization, finite bit representation). For the 2x2 MIMO receiver data was captured at the outputs of the AGC stage, which was then fed to its ideal MATLAB-model counterpart. The resulting raw BER (i.e., no channel coding was included) for both the FPGA prototype and the high-level model, for various SNR values when confronting the same realization of a static Vehicular A channel, can be observed in Figure 5.54a. The same experiment was repeated with a static Pedestrian B channel model, as it can be observed in Figure 5.54b. In both cases it can be seen that real-time implementation achieves nearly identical results with the MATLAB model, demonstrating the robustness and high-precision of the fixed-point FPGA-based implementation.

Similarly, the performance of the FPGA-based receiver was analysed when utilizing both the real-time 2x2 MIMO STBC-based transmitter and its off-line counterpart (i.e., with the MATLAB-generated I/Q vectors being played back by the two VSGs). Figure 5.55 shows the received EVM for both cases under static channel conditions.

The same approach was employed to compare the performance attained with the three different antenna-configurations, as it can be observed in Figure 5.56. In this case the raw BER for the three PHY-layer schemes was obtained considering a static Pedestrian B channel.

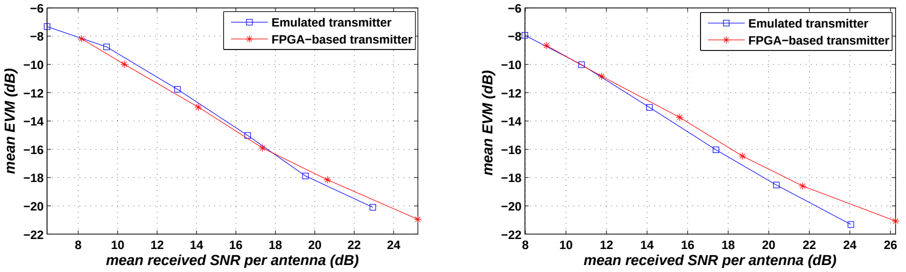
In the previous experimental results, the absence of mobility in the experi-



(a) ITU vehicular A channel

(b) ITU pedestrian B channel

Figure 5.54: Performance-comparison of the 2x2 MIMO FPGA and MATLAB receivers under static channel conditions.



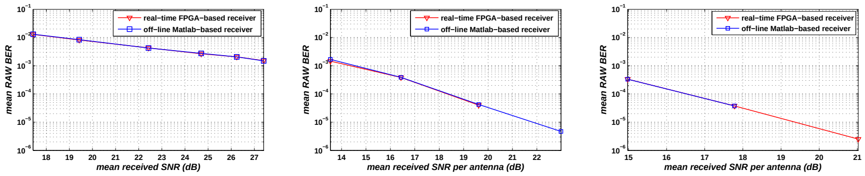
(a) ITU-T vehicular A-based

(b) ITU-T pedestrian B-based

Figure 5.55: Variation in the 2x2 MIMO receiver performance when using the real-time and off-line transmitter prototypes.

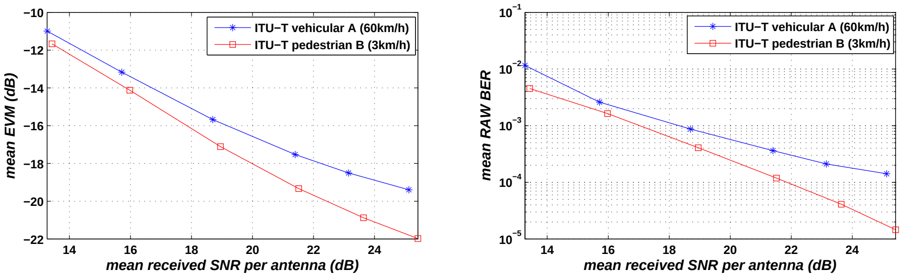
mental testing (i.e., variation of the channel frequency response) is preventing the assessment of the diversity gains introduced by the MIMO scheme. Furthermore, the observed implementation losses are only taking into account a specific channel response. Hence, it is required to account for a broader range of channel conditions to fully assess the system performance and implementation losses. Consequently, an extensive measurement campaign was conducted, requiring of an automated data post-processing stage to extract the BER and EVM metrics. More specifically, the same measurements were repeated for 100 different realizations of the random channel (i.e., employing a different channel seed) and using 6 attenuation steps of the AWGN generators, for each channel realization (using both the vehicular A and pedestrian B models). The mobile velocity for the first case was set at 60 km/h, while a lower speed of 3 km/h was utilized in the pedestrian scenario. Finally, the curves observed in Figure 5.57 were produced by averaging the 100 data captures for each of the 6 attenuation steps. This thorough testing campaign allowed an accurate analysis of the receiver performance under different mobility scenarios.

Moreover, the captures were also utilized in an extensive MATLAB co-simulation (i.e., synthetic-MATLAB signals versus testbed data), which permitted to quantify the implementation losses, which are of approximately 3 dB. The previous figure can be considered acceptable taking into account that the



(a) SISO (b) 1x2 SIMO (c) 2x2 MIMO

Figure 5.56: Performance-comparison of the different antenna-configurations under static-channel conditions.



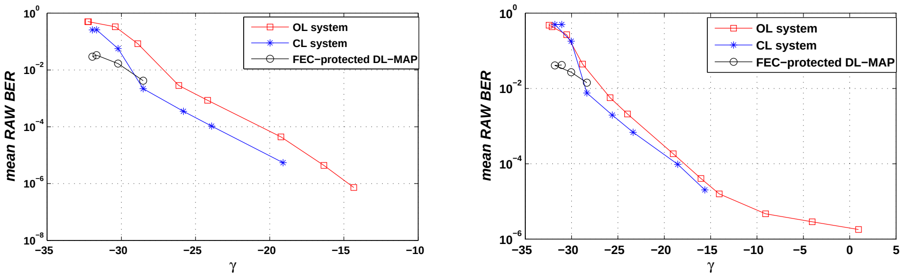
(a) EVM-SNR curves (b) BER-SNR curves

Figure 5.57: Performance of the 2x2 MIMO open-loop prototype in mobile channel scenarios.

entire hardware setup is also contributing to these losses.

The validation of the proposed TAS scheme is also requiring to account for mobile channels. In this case, though, instead of performing an exhaustive measurement campaign as the previously described, the initial performance validation was performed by means of an exhaustive off-line testing approach, which takes advantage of the previously acquired channel captures. Specifically, the RTL-aware MATLAB-model of the system was combined with real-time captures of both a MIMO ITU vehicular A and pedestrian B channels (i.e., the first at 60 km/h and the second at 3 km/h). The utilized captures were providing the uninterrupted channel response of a single time-variant channel realisation over a period of 500 ms (i.e., around 45 complete frames and its associated inter-frame silences).

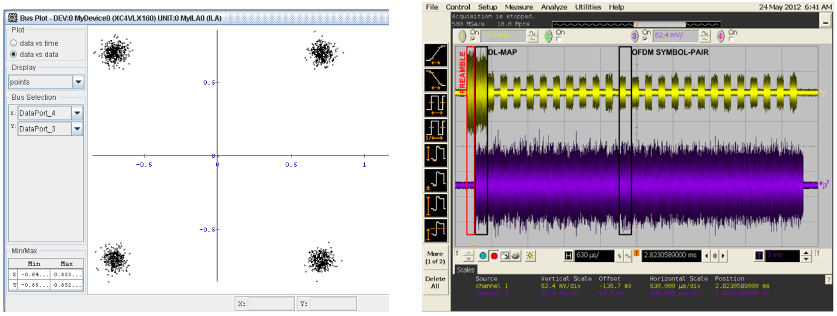
Figures 5.58a and 5.58b show the achieved BER versus γ, where γ is defined as the quotient between the total transmit power and the noise power at the receive antennas. It is important to note that the values of γ should be taken only as relative values, given that the effect of the channel gain is not included. These values allowed the procurement of a fair evaluation of the open-loop (Alamouti-based matrix A STBC) and the TAS-based closed-loop schemes, since the achieved BERs for both cases and for the same total transmit powers can be compared directly. Each point in the curves was obtained as the average of the instantaneous BER over 45 consecutive frames (using the previously described off-line emulation of the single time-variant channel realisation for each channel model). As expected, the TAS communication technique outperforms



(a) ITU pedestrian B (3 Km/h)

(b) ITU vehicular A (60 Km/h)

Figure 5.58: Performance comparison of the implemented MIMO open-loop and closed-loop schemes.



(a) Real-time MIMO-STBC QPSK constellation

(b) Real-time MIMO-TAS IF signal

Figure 5.59: In-laboratory real-time system monitoring.

that of the Alamouti-based system, because the transmitter is able to provide the maximum instantaneous channel gain by allocating all the available transmit power through the selected antenna, whereas in the case of the Alamouti scheme the achieved performance is equal to the mean of the channel gains corresponding to both transmit antennas. Interestingly, the performance of both systems is quite similar for low γ values, because of the errors introduced in the decoding of the DL-MAP information (i.e., impairing the correct operation of the TAS technique).

Finally, it must be stressed out that during the in-laboratory verification it is of extreme importance the utilization of specialized RF instrumentation or embedded FPGA-monitoring logic to analyse in-situ the signals produced at different stages of the real-time prototype. Indicatively, in Figure 5.59a it can be observed a received QPSK constellation via the ChipScope Pro tool, whereas Figure 5.59b shows the real-time MIMO signals at the outputs of the RF downconverters of the receiver antennas where TAS was applied, as visualized through an oscilloscope. For the latter, the instantaneous channel conditions were such that the second transmit antenna was selected to transmit all the the user-data subcarriers. Note, however, that even when this happens, both transmit antennas are used for the transmission of the DL-MAP and the pilot subcarriers according to the standard specifications.

Chapter 6

Use Case II

Realistic implementation and validation of a macrocell/femtocell interference-mitigation technique for LTE-based systems

6.1 Considered system

This chapter provides details of the design, implementation and realistic verification of a macrocell/femtocell interference-mitigation technique targeting high-performance broadband LTE-based wireless communication systems. As in the previous chapter, the focus is laid on the description of the FPGA design and the employed RTL techniques whose goal was to provide a hardware-efficient implementation of adaptive PHY-layer solutions.

6.1.1 Basic introduction to opportunistic frequency reuse

The designated RF spectrum of modern cellular-based wireless communication networks is every time more congested, whilst requiring to serve an increasing number of users (i.e., mobile and fixed). The RF spectrum reuse has been proposed as one of the key technology drivers for the deployment of next generation BWA systems. The efficient deployment of the previous scheme constitutes one of the main goals of CR. However, the opportunistic reuse of the RF spectrum requires the agile mitigation of the effects caused by in-band interfering RF signals. Interference management is therefore becoming an indispensable feature that has to be accounted throughout the joint design of the PHY and MAC layers of network infrastructure equipment, Customer Premises Equipment (CPE) and mobile User Equipment (UE).

Two major interference management categories can be found in the literature. The first one includes interference avoidance techniques such as spectrum sensing, aiming at the instantaneous allocation of unused-unlicensed spectrum. The second one encapsulates interference mitigation schemes that enable high spectral efficiency through frequency reuse, i.e., the same frequency bands are used by different users among adjacent (heterogeneous) cells.

An indicative scenario where interference management schemes are considered a de-facto operational prerequisite is encountered in deployments of 4G wireless access networks, such as the ones defined in the 3GPP LTE standard. In such wireless networks, apart from the standard cellular-like deployment of outdoor BSs, the use of low-power small cellular BSs has gained an increasing popularity in recent years. This is because the existing cellular network operation is facing great difficulties to deliver broadband services to indoor users, due to significant losses provoked by certain construction topologies.

The rise of LTE-based femtocells (i.e., indoor BSs) that cover the needs of short-range residential gateways, largely addresses the indoors topology losses, while their small size optimizes the spatial reuse of radio resources. Indeed, femtocells make feasible the increase of achievable rates per area-unit by reusing the same frequency band assigned to the primary transmission (i.e., between the macro BS and macro UEs). However, the highlighted benefits come at the expense of in-band interference. This is because the simultaneous use of the same operating RF band with equal signal bandwidth can frequently result in Co-Channel Interference (CCI), which in turn could either dramatically deteriorate the performance experienced by the neighbouring macro UEs, or even suspend the provided service. This occurs when the Carrier-to-Interference Ratio (CIR; average received modulated carrier power in relation to the average received CCI power), is below certain critical levels. Thus, better interference rejection capabilities have to be employed during the design of both infrastructure nodes and fixed or mobile equipment.

Interference is hence a major obstacle that can impair the potential gains of small cells, especially when they are deployed in the context of heterogeneous multi-cell networks. For this reason, it is a paramount requirement to apply an adaptive DL transmission among the femto BS and the femto UEs in order to guarantee that no interference is caused to the primary DL communication. This adaptive signal transmission is driven by a suitably selected interference management scheme. In recent years, numerous researchers have proposed techniques to reduce the interference, improve the link-reliability and increase the capacity and performance in macrocell/femtocell scenarios [149]. Moreover, numerous Inter-Cell Interference Coordination (ICIC) schemes have been proposed to serve the operational needs of LTE-based femtocells [150]. These schemes include both interference avoidance techniques [151, 152] and interference mitigation schemes [153, 154].

6.1.2 Short introduction to the 3GPP LTE PHY-layer

The mobile WiMAX and the 3GPP LTE standards (Rel. 9) [LTE, 2010] feature many similarities. Indicatively, as in the case of WiMAX the LTE PHY-layer is based on a scalable OFDMA architecture, with bandwidths ranging from 1.4 to 20 MHz, supporting various MIMO transmission schemes (i.e., Space-Frequency Block Code - SFDM, beamforming and SM). Two CP sizes are defined, where the largest one occupies one quarter of the OFDM symbol (i.e., denoted as extended CP). The 3GPP LTE supports both the TDD and the Frequency Division Multiplexing (FDD) operating modes.

In the FDD mode the OFDM symbols are organized in groups of 12 consecutive subcarriers, known as Physical Resource Blocks (PRBs). The number of PRBs included in each OFDM symbol depends on the bandwidth size (i.e.,

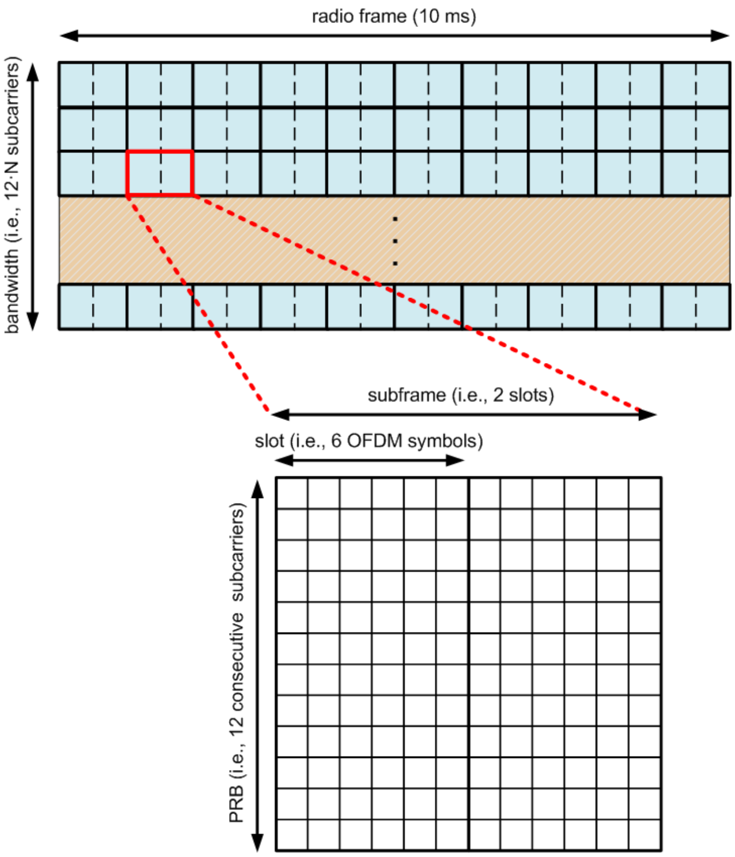

radio frame (10 ms)
bandwidth (i.e., 12·N subcarriers)
subframe (i.e., 2 slots)
slot (i.e., 6 OFDM symbols)
PRB (i.e., 12 consecutive subcarriers)

Figure 6.1: Basic time-frequency FDD LTE-frame structure.

100 in the 20 MHz case). When the extended CP is utilized, each group of six OFDM symbols is named a slot. Furthermore, two slots form a subframe and each frame is composed by ten subframes. The resulting radio frame, shown in Figure 6.1, is 10 ms long.

As in the mobile WiMAX case, the pilot subcarriers, which in the LTE standard are denoted as Reference Signals (RSs), are distributed within the defined slots. The specific RS distribution of the previously presented FDD LTE-frame can be observed in Figure 6.2. As it is shown, RSs are only transmitted in one of each three OFDM symbols. This feature is important, since the channel estimation needs to be calculated using groups of OFDM symbols (i.e., requiring an increased storage capacity).

Additionally, four possible predefined I/Q values can be found for the pilot tones (i.e., $\pm\frac{1}{\sqrt{2}} \pm \frac{1}{\sqrt{2}}i$). Without entering into details, it must be noted that while the location of the RSs is exactly the same in all subframes, two different value-distributions are applied to each half of the frame (i.e., known RS-value sets are defined for each five-subframe set).

Apart from the RSs, the LTE DL frames also include two additional sets of predefined-value subcarriers, namely the Primary Synchronization Signal (PSS) and the Secondary Synchronization Signal (SSS), which facilitate the cell iden-

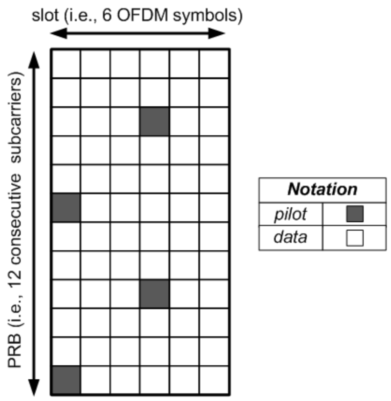

slot (i.e., 6 OFDM symbols)
PRB (i.e., 12 consecutive subcarriers)
Notation
pilot ■
data □

Figure 6.2: RS distribution utilized in the FDD LTE-frame.

tification and synchronization of the UEs. The PSS can be found in the last OFDM symbol of both the first and sixth subframes, while the SSS is always located at the OFDM symbol preceding the one containing the PSS. In an FDD scheme an uninterrupted flow of data is received in the time domain, hence the known location and values of the PSS and SSS allows the identification of the precise subframe that is being received.

6.1.3 Considered scenario and PHY-layer specifications

Designing and implementing an interference management scheme to operate in realistic conditions is a representative situation where the specifications of the underlying analog RF components, signal converters, together with the realization of the DSP algorithms using baseband signal processing boards is an important issue to take into account from the very early stages of development. In fact, the credible validation of this type of algorithms implies the use of a real-time testbed that comprises a series of heterogeneous equipment. Each of the underlying hardware components has an impact on the performance of the developed algorithms. Whereas essential, the computer-based modelling can only be considered as a first step towards the real-life implementation and performance validation of a complex system. The interference mitigation in LTE-based HetNets is a typical case where the real-life implementation of the PHY-layer using a dedicated real-time baseband processing solution, accounts for the low-level hardware specifications and limitations of the target platform (digital and analog), the signal impairments, the features of the propagation channel and other physical constraints; such factors and conditions are not always considered or thoroughly emulated when modelling the same PHY-layer using a high-level computer-based simulation. For this reason, the development of such algorithms should naturally target a proof-of-concept implementation of the PHY-layer using close to real-life operating scenarios. Among a variety of pragmatic considerations, real-time PHY-layer implementations like the one that is presented in this chapter, take into account the translation of the floating-point arithmetic of the initial high-level system model to a fixed-point one, the finite processing resources and capacity of the selected real-time baseband processing solution and the high impact on the overall development complexity

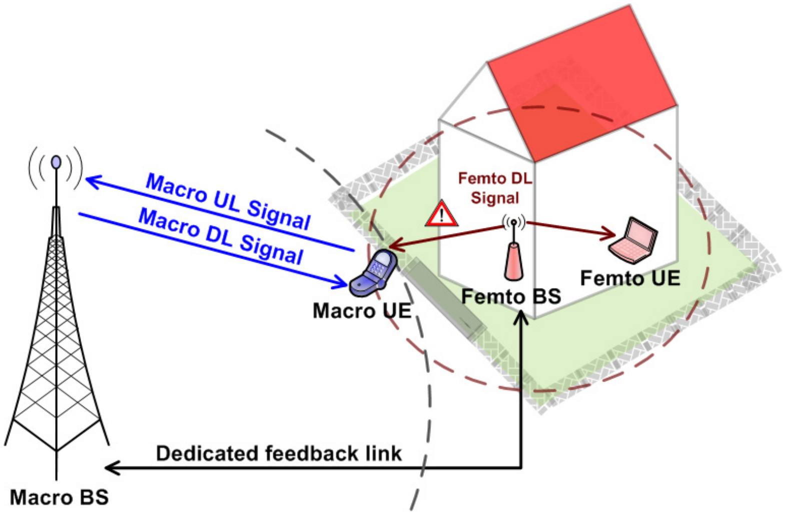


Figure 6.3: Considered LTE-based macrocell/femtocell frequency reuse scenario.

of the low-level control plane design (i.e., especially when it is required high run-time adaptivity of the PHY-layer building blocks).

When combining wide channel bandwidth (e.g., 20 MHz or more) and adaptive baseband algorithms whose performance or operability heavily depends on the instantaneous channel response at the mobile receivers, it is fair to claim that current computer-based simulations are struggling to cope with the required run-time processing load. Implementing the mentioned systems in specialised signal processing platforms resolves the performance and validation issues. Nevertheless, a wide and densely populated channel bandwidth brings additional considerations when implementing a real-time system at a chosen baseband signal processing platform (i.e., storage of large chunks of data, fast access of data, control plane with high complexity and interdependencies, bit-intensive signal processing at baseband).

To illustrate the necessity of DL interference management strategies, the scenario depicted in Figure 6.3 has been considered, where the primary DL communication between a given macro BS and a macro UE is receiving interference from a secondary DL communication between a femto BS and a femto UE. The femto BS applies an opportunistic transmission at the same RF band and with the same bandwidth as the primary macro communication. For the sake of simplicity, the femto UL signal is not depicted in the figure.

As it can be observed in Figure 6.3, the macro UE is located near the cell edge and near the femto BS, hence becoming a potential victim of in-band interference caused by the secondary transmission. If no interference management strategy is implemented and applied at this system, the quality of the signal received by the macro UE may significantly degrade, and thus negatively affect the quality of service perceived by the end user (macro UE). To avoid this undesirable situation an interference mitigation technique for the femto BS is put in place, based on a self-organized PRB allocation scheme. More specifically, if the macro UE detects interference in the DL signal, it notifies its associated BS (through a dedicated feedback link) which, in turn, requests the femto BS to adapt its transmission in order not to interfere the primary communication. The presented use case provides a baseline validation of this interference management scheme demonstrating the benefits in a practical and tangible way.

Both case studies described in this thesis focus on point-to-point DL communications. In the case study presented herein, both the UL communication between the macro UE and the macro BS and the one between the macro BS and the femto BS are emulated. As in the first use case, only a subset of the PHY-layer features described in the LTE standard were implemented. Nevertheless, the considered DL OFDM frame is respecting the format described for the FDD operation mode defined by the LTE standard. It is important to underline that a high bandwidth of 20 MHz was selected, which as discussed earlier is influencing the amount of required processing resources (and thus depending on the target FPGA device, it may influence as well key design and implementation decisions). Table 6.1 summarizes the PHY-layer specifications.

Parameter	Value
Wireless telecommunication standard	3GPP LTE (Rel. 9)
Antenna scheme: SISO	1x1
Channel bandwidth (MHz)	20
Cyclic prefix (samples)	512 (1/4 of the symbol)
Modulation type	QPSK
Duplex mode	FDD
Active subcarriers per OFDM symbol	1200
Null subcarriers per OFDM symbol	848
FFT size	2048
OFDM symbols per frame: total ||active	120 ||83
Closed-loop transmission scheme	Interference-aware PRB allocation
ADC sampling frequency (MHz)	61.44
Baseband sampling frequency (MHz)	30.72
RF band (GHz)	2.6
IF (MHz)	46.08
Tested channel model	ITU Ped. B (up to 3 km/h)

Table 6.1: Considered LTE PHY-layer specifications.

In more detail, the LTE-based frames were formatted in a suitable way to serve the proof-of-concept needs of the macrocell/femtocell interference-management mechanism, comprising periods where no data-transmission occurs. This implies that the UE knows a priori that no PRBs are allocated in given zones of the frame (i.e., non-active OFDM symbols). In the primary communication, all the PRBs in the active OFDM symbols are dedicated to the macro UE. The macro UE feedback defines which PRBs must be activated for the femto UE in the opportunistic femto transmission scheme. The RSs are included in all the OFDM symbols in the frame (i.e., active and non-active). The resulting quasi-quiet periods facilitate vital DSP operations implemented by the DFE of the receiver, such as the correction of the CFO or the AGC gain-modification. Figure 6.4 shows the utilized frame structure, where it can be observed that two different 5ms-frames are transmitted in the 10 ms period defined in the LTE standard.

Although the PSS and SSS structures are included in the transmitted considered LTE-frame, a simpler synchronization method has been employed based

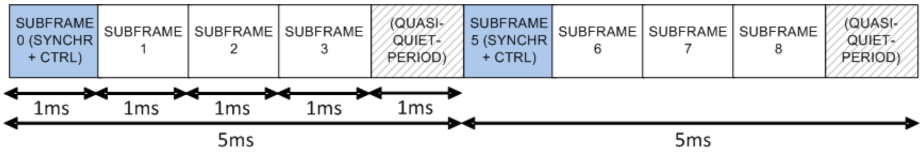


Figure 6.4: Specific frame format utilized in the LTE-based system.

on the symbol repetition introduced in the DL signal by the CP.

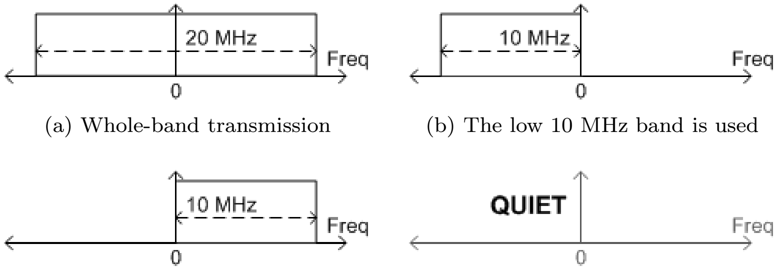
The pathloss model for suburban deployment of LTE femtocells, defined by the 3GPP [(3GPP), 2009], was adopted to define the range of acceptable Signal-to-Interference Ratio (SIR) values. In order to achieve this, it was necessary to model the pathloss of the three DL signals. Hence this model included the DL communication between the macro BS and the macro UE, the equivalent femto DL communication and finally the interfering signal between the femto BS and the macro UE. It has to be noted that no interference was assumed for the femto DL communication due to the macro BS transmission. Therefore, in the proposed proof-of-concept, the SIR is defined as the ratio between the signal generated by the macro BS and the interfering signal. Apart from the pathloss model, other parameters such as the macro UE speed, the inter-site distance and the house size were also taken into account in order to finally define a SIR level between 12 and 20 dB for our considered testing scheme.

Despite the fixed PHY-layer parameters and the simplified FDD transmission scheme, the developed system is a complete real-time baseband prototype which accounts for real-life channel propagation and interference conditions.

6.1.4 Interference management scheme

A novel distributed ICIC technique, known as Victim User Aware Soft Frequency Reuse in macrocell/femtocell HetNets [Shariat et al., 2012], was selected to realize the proof-of-concept interference management scheme. The main objective of this technique is to protect the macro UEs from the DL transmission of a neighbouring femto BS. In a frequency reuse scenario where both macro and femto communications utilize the same RF band, the available bandwidth is divided in various subbands. In order to achieve dynamic interference mitigation, the instantaneous channel condition of the macro UE (victim) is exploited to adapt the femto transmission (i.e., to avoid interfering the primary communication). A basic objective is to deactivate the lowest number of required subbands in the secondary femto communication. Thus, the proposed technique aims at improving the throughput of the macro UE, while minimizing the impact on the performance of the femtocell.

A downscaled version of the previously described interference management scheme was adopted, in order to reduce the implementation complexity without compromising the proof-of-concept goals; this featured a single macro BS/UE pair and one femto BS/UE pair. In this proof-of-concept case, the macro BS DL transmission uses the whole 20 MHz bandwidth, whereas the femto BS DL transmission uses two predefined 10 MHz bands. As it may be observed in Figure 6.5 this results in four different femto DL transmission-schemes (i.e., adaptive PRB allocation).



(a) Whole-band transmission

(b) The low 10 MHz band is used

(c) The high 10 MHz band is used

(d) The femto BS remains quiet

Figure 6.5: Predefined femto PRB allocation cases.

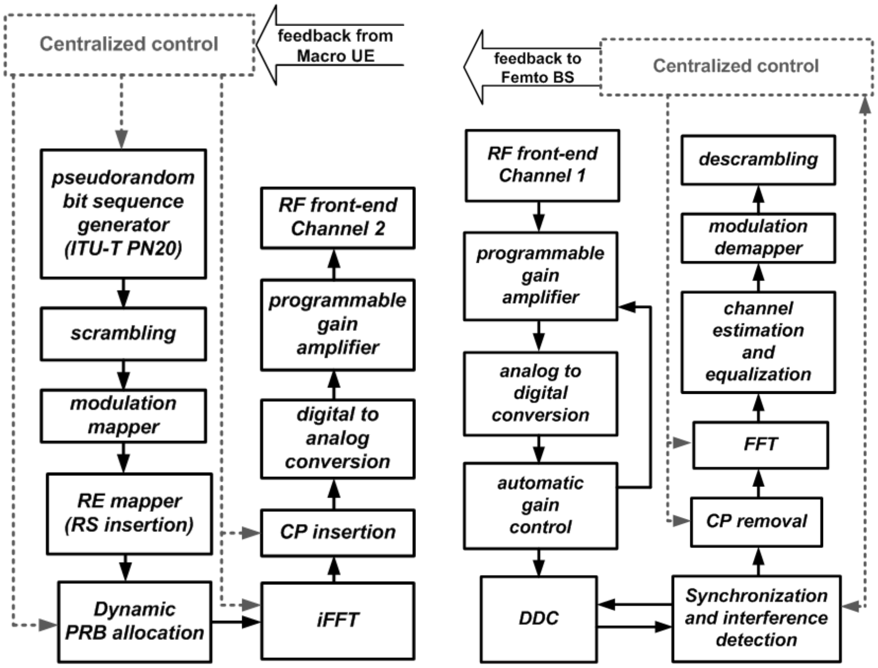
As previously mentioned, the macro BS and femto BS DL signals use the same RF band; hence, if both signals occupy the complete 20 MHz bandwidth and the mobile channel between the femto BS and the macro UE is not featuring deep fading in any of the defined 10 MHz bands, then the interference caused by the femto BS would prevent or severely impair the DL communication between the macro BS and the macro UE. For this reason, the macro UE requests at run-time a different femto BS transmission-scheme. In more details, the PRB allocation of the femto BS is dynamically adapted according to the performance requirements of the macro UE, which are subject to the instantaneous channel conditions. The starting point of the interference management scheme is encountered in the macro UE, which executes an interference-detection algorithm (more details are given in Section 6.2.1) that determines if there is presence of interference or not. On top of that, the algorithm specifies the band where interference is detected. Using that information, a feedback is generated defining which of the four femto BS transmission modes is going to be used. The interference management decision tries to guarantee the best femto DL communication. Hence, the no-transmission scheme is forced only when the interference is detected in the complete 20 MHz bandwidth. Finally, the femto UE is able to follow the adaptive PRB allocation.

6.2 Utilizing an incremental development

The interference-mitigation scheme presented in this chapter can be considered an evolution of the single-antenna system presented in Section 5.2.1. Although the underlying wireless communication standard is different, part of the RTL design presented in the previous chapter has been reutilized. To start with, the model of the signal received at the macro UE features a great similarity with the one presented in (5.1):

$$\begin{aligned} c(t) =&\Re\{x(t)\cdot e^{j2\pi(f_{IF}+\Delta f)t}\} + \Re\{u(t)\cdot e^{j2\pi(f_{IF}+\Delta f_u)t}\} + A \\ &+ B\cdot cos(2\pi(f_{IF}+\Delta f)t+\varphi) + w(t), \end{aligned} \tag{6.1}$$

where $x(t)$ represents the useful part of the received baseband signal, $u(t)$ is the interference, f_{IF} is the IF, Δf and Δf_u is the CFO for the desired and interfering signals respectively, A is the DC level introduced by the baseband board chassis, $B\cdot cos(2\pi(f_{IF}+\Delta f)t+\varphi)$ represents the unwanted residual carrier, located at



(a) Femto BS (b) Macro UE

Figure 6.6: General block-diagram of the implemented LTE-based PHY-layer.

the center of the useful signal-spectrum (i.e., introduced by the LO coupling at the transmitter) and finally, $w(t)$ is the zero-mean white circularly symmetric Gaussian noise. The received baseband signals can be expressed as follows:

$$x(t) = \tilde{x}(t) * H(t), \tag{6.2}$$

$$u(t) = \tilde{u}(t) * H_u(t), \tag{6.3}$$

where $\tilde{x}(t)$ is the equivalent transmitted baseband signal, $\tilde{u}(t)$ represents the equivalent interfering baseband signal and $H(t)$ and $H_u(t)$ are the equivalent baseband representations of the corresponding channel responses for the desired and interfering signals, in respect to the center RF frequencies, $f_{RF} + \Delta f$ and $f_{RF} + \Delta f_u$, of each of the two signals.

The focus of this chapter is on the design of an extended DFE, which provides the core functionalities of the interference-mitigation scheme. The baseband processing stages of the femto BS and macro UE are shown in Figure 6.6. Taking advantage of the incremental development principles, the re-utilized building blocks are listed hereafter:

(i) Certain processing blocks have been directly reused (i.e., symbol mapping/de-mapping, insertion and removal of the CP and FFT/IFFT). Similarly, the PRBS generator utilized at the macro BS is directly reused from the WiMAX system, whereas a variation of the same logic, based on the ITU PN20 specification [(ITU), 1996], has been selected for the femto BS.

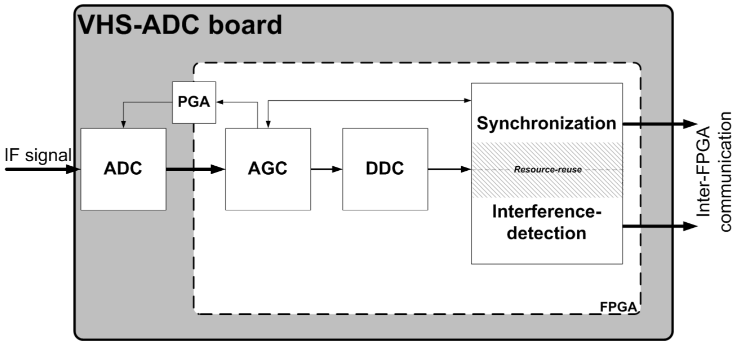


Figure 6.7: Extended DFE architecture of the macro UE.

(ii) The LTE-based system also takes advantage of the channel estimation processing architecture that was presented for the WiMAX receiver. For instance, the speed-optimized pipelined DSP structure that calculates the quadratic interpolation was directly reused. At he same time the internal memory system and neighbouring pilot-set selection algorithmic had to be slightly modified considering that the channel estimation in LTE is performed for each group of three OFDM symbols (in WiMAX the channel was estimated using OFDM symbol pairs) and according to the defined RS distribution.

(iii) The core DFE functionality is an adapted version of the one presented in the first use case. The modifications are encountered in the AGC, which takes into account the longer frame format, and also in the DDC and synchronization processing stages, which were configured according to the LTE system specifications (recall Table 6.1). As already mentioned, the DFE of the macro UE was extended to provide the required interference-management features, as it is detailed in the following section.

6.2.1 Operation of the extended DFE

The DFE plays a critical role in the correct operation of any real-life receiver and greatly influences the overall performance of the system. As already detailed in Section 5.2.1, the DFE comprises the AGC, DDC and synchronization processing blocks. The role of the AGC and DDC is to exploit the full dynamic range of the received signal and provide the baseband signal respectively (i.e., sampled at 30.72 MHz as defined by the LTE standard); the synchronization on the other hand, locates the FFT window of each OFDM symbol, estimates the CFO (which is used to re-tune the DDS of the DDC) and controls the operation of the AGC.

On top of these core DFE functionalities, the macro UE needs to detect in a timely manner any interference that might originate from the femto BS DL transmission, since its presence might prevent the decoding of the received signal. Thus, the functionality of the DFE has been extended to include interference detection capabilities which were also complemented with feedback generation and signalling features. A high-level representation of the proposed interference-aware DFE is presented in Figure 6.7. Taking into account that this design has to be hosted in a single Virtex-4 LX160 FPGA device, it was required to employ a rigorous resource-reuse strategy at algorithmic level.

Interference-detection algorithm: design decisions for a resource-optimized RTL design

Taking into account the defined scenario, the proposed interference-mitigation scheme relies on a joint implementation of the synchronization and interference-detection mechanisms. The symbol synchronization (i.e., location of the FFT window) was designed exploiting the self-similarity of the received OFDM symbols due to the presence of a CP at the end of each symbol. Likewise it was achieved a reutilization of the synchronization technique presented in Section 5.3.1. Moreover, it was avoided to implement the far more complex synchronization method that is based on the utilization of the SSS/PSS signals of the LTE-based DL frame (hence, saving useful FPGA resources).

An algorithm optimization, favouring a minimization of the required computation resources, is applied to achieve an interference-aware synchronization technique. Specifically, as in the case of the WiMAX receiver, a cross-correlation is performed to detect the CP and locate the FFT window of the samples comprising each OFDM symbol. Furthermore, the values of the cross-correlation can be opportunistically reused to detect the presence of a non-negligible interference in the DL signal, relying on the fact that the degradation of the cross-correlation utilized to locate the CP is directly related to the SIR conditions at the receiver (i.e., the peak values of the correlation decrease in the presence of an interfering femto DL signal).

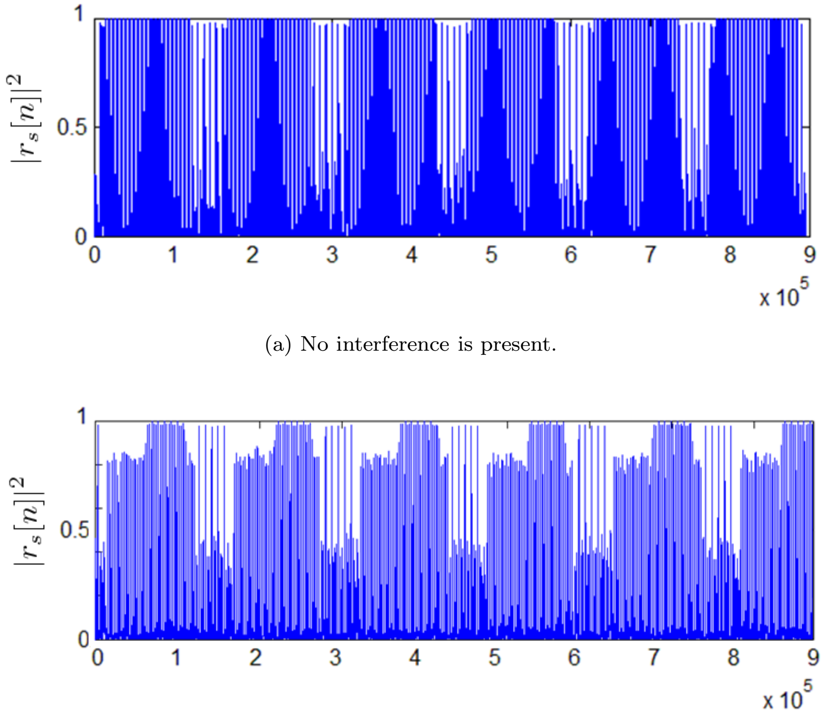
When the interferer's CP is aligned in time with the CP of the received user-data frames, the interference is denoted as synchronous, whereas when the CPs are not aligned, it is denoted as asynchronous. In the considered system, given the nature of the proposed interference-detection mechanism, it is assumed that the interferer is asynchronous.

The calculation of the cross-correlation follows the same algorithmic rationale of the first use case, which was appropriately adapted to meet the specifications of the considered scenario. More specifically, given the impulse response of the considered ITU Pedestrian B channel model, the sliding window utilized in the main processing branch was resized to 2048+467 samples (i.e., 45 samples were discarded due to the effect of the mobile channel). The expression corresponding to the square of the correlation when the sliding window starts at the nth sample is given by:

$$|r_s[n]|^2 = \frac{|\sum_{l=0}^{466} s^*[n+l] \cdot s[n+l+2048]|^2}{(\sum_{l=0}^{466} |s[n+l]|^2) \cdot (\sum_{l=0}^{466} |s[n+l+2048]|^2)}, \qquad (6.4)$$

where $s[n]$ is the equivalent complex baseband signal at the output of the DDC, sampled at 30.72 MHz.

A peak in $|r_s[n]|^2$, indicates the detection of the symbol and thus the sample where the CP starts. In a scenario without noise and without interference, the amplitude of the peaks of $|r_s[n]|^2$ is almost equal to one. Nevertheless, in the presence of noise and interference the cross-correlation profile is degraded and the amplitude of the peaks decreases, as it can be observed in Figure 6.8. The amount of degradation is directly related with the power of the received interference. Hence, the presence of interference can be detected by accurately defining



(a) No interference is present.

(b) SIR of 12 dB.

Figure 6.8: Degradation of the cross-correlation profile in the presence of interference.

a threshold value (i.e., a peak value below the defined threshold indicates the presence of interference).

Figure 6.9 shows an overview of the baseband processing architecture that was designed towards this end. The interference detection scheme is based on three signal processing branches. The main branch jointly implements the synchronization and the interference-detection mechanism, which scans the entire 20 MHz band to identify the presence of interference. The other two branches detect interference on two predefined 10 MHz subbands. These half-band interference detectors are built with the help of precise low-pass and high-pass complex filters (i.e., a complex-valued FIR stage is applied to the baseband signal produced by the DDC, in order to separate each defined 10 MHz band). Furthermore, a latency-compensation FIFO memory (dimensioned according to the latency of the digital filtering stage) ensures the time-alignment of the three interference-related DSP branches. The proposed interference detection algorithm tracks the effect of interference on the normalized CP correlation. For this reason it was necessary to define a threshold in this correlation; if the minimum correlation peak of a given data period is below this threshold, then interference is detected. In order to decide if the interference is present in the whole bandwidth, or in the upper or lower half of the band, Algorithm 2 is applied at each 5 ms-frame.

The three processing branches are producing the inputs of the feedback

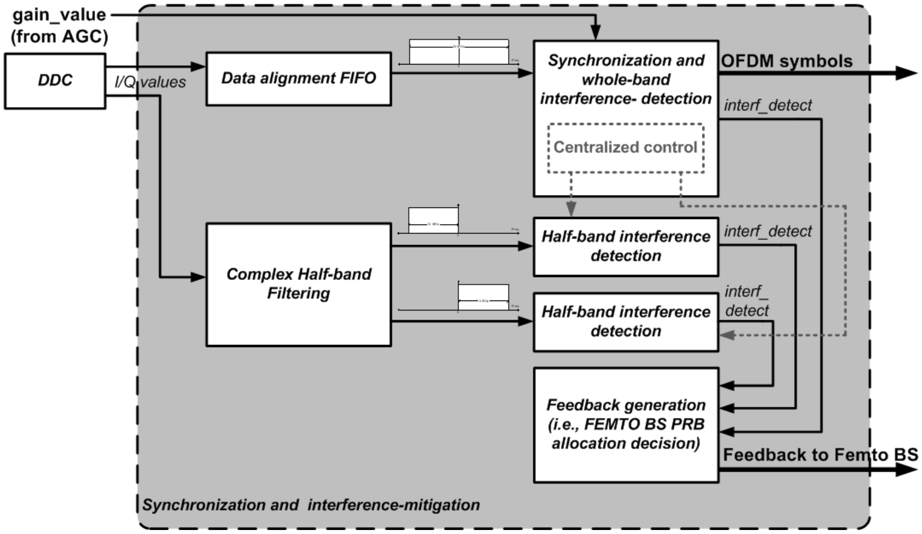

gain_value
(from AGC)
DDC
I/Q values
Data alignment FIFO
Synchronization and whole-band interference- detection
Centralized control
OFDM symbols
interf_detect
Complex Half-band Filtering
Half-band interference detection
interf_detect
Half-band interference detection
interf_
detect
Feedback generation (i.e., FEMTO BS PRB allocation decision)
Feedback to Femto BS
Synchronization and interference-mitigation

Figure 6.9: Joint synchronization and interference-detection design.

Algorithm 2

```
if wholeband_detection == 0 then
    decision = no interference;
else
    if low_10MHz_band_detection == 1 and high_10MHz_band_detection == 0 then
        decision = interference detected in the low 10 MHz band;
    else if low_10MHz_band_detection == 0 and high_10MHz_band_detection == 1
    then
        decision = interference detected in the high 10 MHz band;
    else
        decision = interference detected in the entire bandwidth;
    end if
end if
```

generation block, which is reduced to a basic logic table, that provides a two-bit feedback signal (i.e., 00 = no interference, 01 = interference in the low 10 MHz band, 10 = interference in the high 10 MHz band and 11 = whole-band interference). Finally, the architecture is completed by a centralized unit (found in the whole-band detection branch) which implements the control-plane of the proposed PHY-layer scheme.

The objective of the defined thresholds was to guarantee a minimum performance-level for the macro UE. Upon interference detection, the threshold-values must fulfil the following Key Performance Indicator (KPI): the probability that the raw/uncoded BER is below 10^{-2} must be above 0.8 (conditioned on the fact that interference is detected).

In order to define the optimum threshold values, extensive simulations were conducted in Matlab utilizing synthetic test vectors (i.e., also produced in Matlab), or based on realistic signals that were captured with the help of the GEDOMIS® testbed. More specifically, MATLAB-produced vectors for both the macro and femto BSs were loaded into two VSGs that produced the RF signals of interest. These were fed to the channel emulator allowing the real-time emulation of the selected channel model and the precise control of the produced interference (e.g., affected bands). The resulting macro UE DL signal was then downconverted to an IF and finally sampled and captured on the VHS-ADC prototyping board (see Figure 6.7). This process was repeated for different channel configurations and the captured data were post-processed in MATLAB using the high-level model of the extended macro UE DFE. As a result, it was selected $Th_{\text{whole band}} = 0.93$ for the whole-band interference-detection and $Th_{10 \text{ MHz band}} = 0.90$ for the high and low 10 MHz-band interference detection (more details for the selection of the interference detection thresholds are given in [Font-Bach 13a]). The selected thresholds permit to achieve an optimum trade-off between the probability of detection and the mean attained BER.

6.3 Efficient RTL design of the complete communication scenario

While the proposed interference-management scheme might seem simple at algorithmic level, its realistic validation demands a complex digital realization that is heavily conditioned by the challenging PHY-layer features (i.e., large bandwidth, closed-loop communication, adaptive PRB allocation and interference-mitigation capabilities) and the computationally intensive DSP functions. In fact, the joint synchronization and interference-detection/mitigation technique constitutes one of the most complex processing stages in the considered LTE-based system, since their building blocks result in a resource-hungry processing stage, featuring a number of complex FIR filters, a phase extraction function, one division and a number of multiplications among other arithmetic operations. Therefore, its real-time implementation requires a hardware-efficient RTL design which in fact is the main innovation presented in this chapter.

In the following subsections it is detailed the RTL design of the digital filtering stage, the joint synchronization and wholeband-detection technique and the centralized control unit.

6.3.1 Hardware-efficient implementation of a complex digital filtering stage

The MATLAB Filter Design and Analysis Tool (FDATool) was used to design the required low-pass and high-pass filters of the half-band interference-detection branches (recall Figure 6.9). The FDATool produces coefficients in a file-format tailored for the Xilinx FIR IP core. The FIR filter architecture accounts for a complex input signal; hence, the resulting filter coefficient-sets are also complex-valued. Taking into account that the Xilinx FIR IP core only accepts real-valued coefficients, two instances are required in order to implement each complex filter (i.e., one for the real part, $h_i[n]$, and another for the imaginary part, $h_q[n]$). The filters have been designed satisfying a trade-off between the number of coefficients of the filter and the out-of-band rejection. This trade-off also accounted for the available FPGA resources of the target platform. A suitable set of 51 18-bit complex coefficients was selected exhibiting an attenuation of 35 dB in the rejection band. The duplication of the FIR filters was an important design

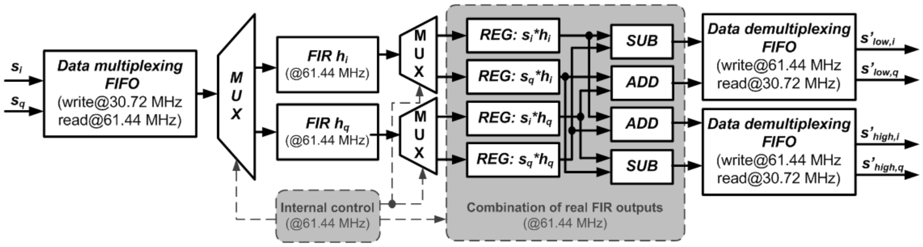


Figure 6.10: Proposed time-shared architecture for the complex FIR filters.

concern, since a combination of filters with such specifications consumes a large amount of FPGA-resources (DSP48 or generic slices depending on the underlying implementation particulars). For this reason resource sharing techniques have been employed in order to tackle the limited capacity of the FPGA device in the target validation platform.

An efficient way to reduce the implementation complexity was to design filters with symmetric response (featuring an even-symmetric coefficient set). Furthermore, the coefficients of the FIR filter that isolates the high 10 MHz band, $h_{\text{high}}[n]$, are the complex conjugate of its low 10 MHz band counterpart, $h_{\text{low}}[n]$, as indicated hereafter:

$$h_{\text{low}}[n] = h_i[n] + j \cdot h_q[n], \tag{6.5}$$

$$h_{\text{high}}[n] = h_i[n] - j \cdot h_q[n]. \tag{6.6}$$

The resulting RTL design is taking advantage of the previous characteristic as shown in Figure 6.10. The two complex filters were implemented using only two FIR filter instances by exploiting a resource-sharing architecture. A data-multiplexing technique is allowing this portion of the system to work at two-times the baseband frequency (i.e., 61.44 MHz). The FIR filters process real signals and feature two internal processing channels; this fact allows them to process concurrently two different input sample streams (i.e., the real and imaginary components of the produced I/Q data-stream are processed separately in an interleaved fashion). Whereas a new I/Q value is available each 32.55 ns, the two-channel FIR instances are requiring a new real-valued input each 16.28 ns. During the first half of each 32.55 ns time-slot, the real component (i.e., first channel) of the incoming DDC output is delivered to the two filter instances. Similarly, its imaginary part (i.e., second channel) is introduced to the two filters during the second half of the 32.55 ns time-slot. In order to implement this solution, dedicated switches and FIFO memories with independent read and write clocks are utilized to provide a reliable cross-clock domain data communication. The produced filter-outputs are then demultiplexed to produce the complex filter outputs at the baseband frequency of, 30.72 MHz. In more detail, the real and imaginary operands of the incoming samples are processed sequentially in a custom pipelined architecture, as detailed hereafter:

(i) The incoming I and Q operands, $s_i[n]$ and $s_q[n]$, are stored at 30.72 MHz.

(ii) $s_i[n]$ is read at 61.44 MHz in the first processing channel of the filter.

(iii) The convolution of this operand with the real and imaginary operands of the filter coefficients is calculated (i.e., $s_i[n] * h_i[n]$ and $s_i[n] * h_q[n]$). In parallel $s_q[n]$ is read at 61.44 MHz in the second processing channel of the filter.

(iv) The outputs of the first channel of the filter are stored at 61.44 MHz. In parallel, the equivalent calculations are repeated for $s_q[n]$ (i.e., $s_q[n] * h_i[n]$ and $s_q[n] * h_q[n]$).

(v) The outputs of the second channel of the filter are stored at 61.44 MHz.

(vi) The output-values of the complex filters, $s'_{\text{low}}[n] = s[n] * h_{\text{low}}[n]$ and $s'_{\text{high}}[n] = s[n] * h_{\text{high}}[n]$, are calculated as follows:

$$s'_{\text{low},i}[n] = s_i[n] * h_i[n] - s_q[n] * h_q[n] \tag{6.7}$$

$$s'_{\text{low},q}[n] = s_q[n] * h_i[n] + s_i[n] * h_q[n] \tag{6.8}$$

$$s'_{\text{high},i}[n] = s_i[n] * h_i[n] + s_q[n] * h_q[n] \tag{6.9}$$

$$s'_{\text{high},q}[n] = s_q[n] * h_i[n] - s_i[n] * h_q[n] \tag{6.10}$$

The detailed operations require two 16.28 ns time-slots (i.e., two-channel filters operating at 61.44 MHz).

(vi) The resulting output samples are stored at 61.44 MHz.

(vii) The filtered samples are read at 30.72 MHz and forwarded to the half-band interference detection components.

6.3.2 Joint design of the synchronization and the interference-detection

As it has been introduced previously, it is proposed a design favouring a minimization of the required computation resources, to achieve an interference-aware synchronization technique. The latter is based on the calculation of a cross-correlation to detect the CP heading each OFDM symbol. The reduced complexity RTL design presented for the WiMAX system was reutilized, in order to attain a minimized arithmetic-implementation cost of (6.4), as indicated hereafter:

$$|r_s[n]|^2 = \frac{|dn[n]|^2}{ds0[n] \cdot ds1[n]}, \tag{6.11}$$

where the elements in the numerator and denominator are calculated in a recursive way:

$$dn[n+1] = \begin{cases} dn[n] + s^*[n+467] \cdot s[n+2048+467] \\ \qquad \text{if } n \leq 467 \\ dn[n] - s^*[n] \cdot s[n+2048] \\ \quad + s^*[n+467] \cdot s[n+2048+467] \\ \qquad \text{if } n > 467, \end{cases} \tag{6.12}$$

where $s[n]$ is the equivalent complex baseband signal at the output of the DDC, sampled at 30.72 MHz, and with $dn[0] = 0$ ($ds0[n]$, $ds1[n]$ are calculated in a similar manner). With this optimization only four samples need to be introduced to the already calculated correlation, thus resulting in a much less DSP-block

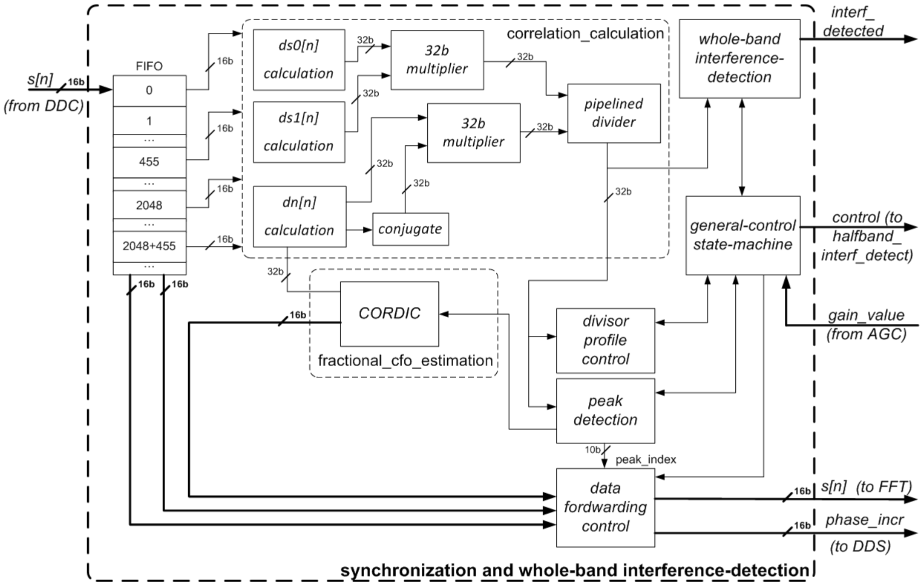


Figure 6.11: Detailed RTL design of the synchronization and wholeband interference-detection block.

demanding FPGA realization. A reduced version of (6.11) is also implemented in each half-band interference-detection processing branch; taking into account the length of the impulse response of the complex filters, the sliding window is using 51 samples less.

The proposed RTL architecture for the joint synchronization and whole-band interference-detection technique extends the one utilized in the mobile WiMAX use case, as shown in Figure 6.11. It must be noted that the threshold-based peak detection mechanism has been adapted to the LTE signal format. Specifically, the transmission of the RSs produces peak-values during the quasi-quiet periods, when ideally a peak value in the correlation should indicate the location of the CP of an OFDM symbol containing user-data. Hence, to determine whether a correlation peak is located within a user-data period or not, the value of the divisor from (6.11) is used, since $ds0[n] \cdot ds1[n]$ is naturally higher during the user-data periods (as shown in Figure 6.12). For this reason, a control process keeps track of the peak-values of the divisor, accounting for the gain-variations applied by the AGC. Finally, it must be noted that the internally utilized registers are reset at the beginning of a newly detected 5ms-frame to avoid problems with the accumulated arithmetic-error due to the finite precision of the fixed-point representation.

A dedicated state machine is in charge of implementing the whole-band interference detection, based on the values of the cross-correlation utilized for the symbol detection and the previously defined whole-band interference-detection threshold, $Th_{\text{whole band}}$. Further details are provided in the following section.

The half-band interference-detection processing blocks are implementing a reduced version of the previously described architecture. This includes the calculation of the cross-correlation (using a sliding window of 416 samples), with the associated peak-detection and divisor profiling logic, and also a dedicated

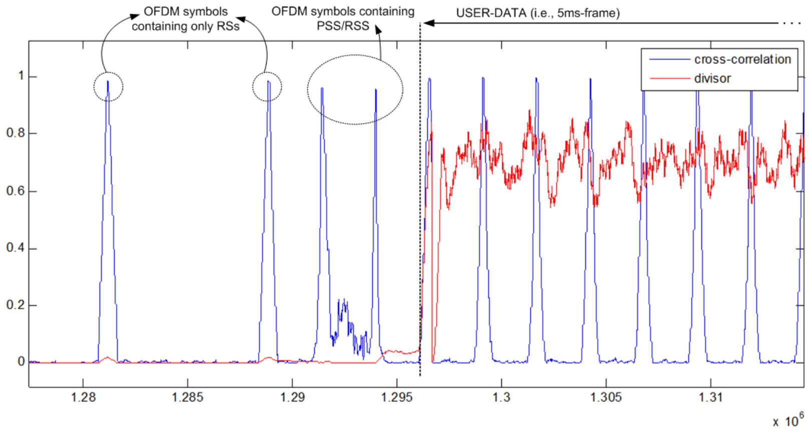


Figure 6.12: Cross-correlation and divisor values when a 5ms-frame is detected.

half-band interference-detection state machine.

6.3.3 Centralized control unit

It is important to underline that the selected synchronization technique allows a low-complexity implementation of the interference-detection scheme detailed in Section 6.2.1. Nonetheless, the large number of concurrent DSP operations require a carefully designed timing control of the diverse data-path stages. For this reason, a centralized controller was implemented to manage the synchronous operation of the above described sub-processes. This centralized control unit is also responsible for triggering the operation of the interference-detection DSP branches. In order to achieve resource-reuse the DFE and its interference detection extension have been jointly addressed at design time, yielding an efficient FPGA implementation. This was achieved by utilizing a hierarchical structure of dedicated state machines.

Figure 6.13 details the underlying state machine implementing the core functionalities of the centralized control unit of the macro UE. This unit tracks the symbol detection process, based on the values of the calculated cross-correlation, taking into account as well the values of its divisor. Likewise, it is detected the CP of the OFDM symbol including the PSS/SSS symbols of each 5ms-frame (i.e., FFT-window location). The symbol detection is then triggering both the operation of the interference-detection sub-blocks and the forwarding of data to the remaining baseband processing stages of the receiver. For the sake of clarity, the figure is not including the management of the AGC gain-variations (i.e., critically affecting the utilized divisor-threshold value). Similarly, the FFT/CP control processes which are also activated with the symbol detection logic have not been included.

In Figure 6.13 it can also be observed the dedicated state machine used to control the whole-band interference-detection logic. Basically, if a single peak-value within a 5ms-frame is below the defined threshold, then it will be indicated that interference has been detected in the main branch (recall Algorithm 2). The dedicated half-band interference-detection state machines present a similar structure and are managed by the centralized controller in an equivalent way.

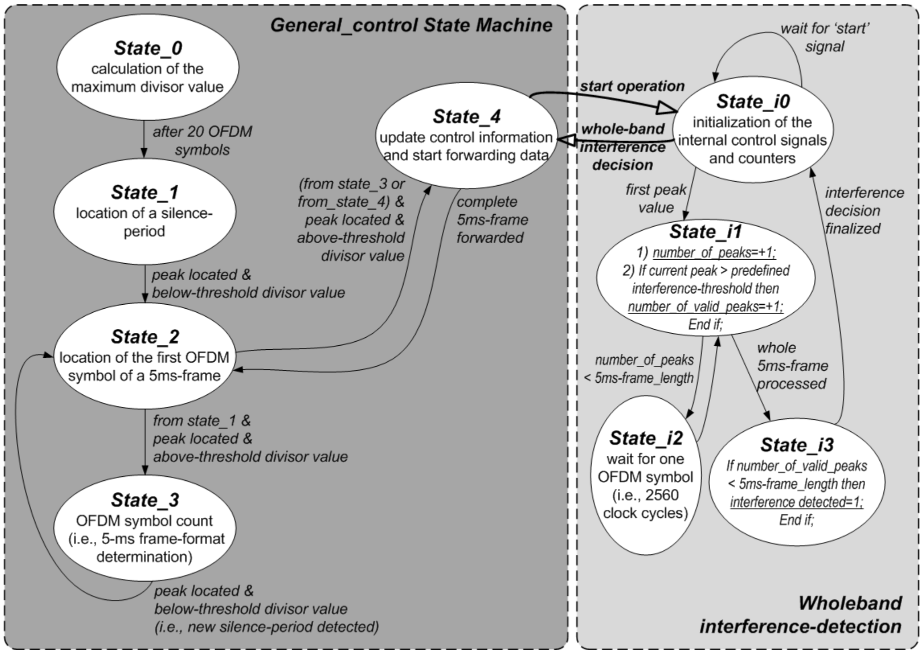

General_control State Machine
State_0
calculation of the maximum divisor value
after 20 OFDM symbols
State_1
location of a silence-period
peak located & below-threshold divisor value
State_2
location of the first OFDM symbol of a 5ms-frame
from state_1 & peak located & above-threshold divisor value
State_3
OFDM symbol count (i.e., 5-ms frame-format determination)
peak located & below-threshold divisor value (i.e., new silence-period detected)
State_4
update control information and start forwarding data
(from state_3 or from_state_4) & peak located & above-threshold divisor value
complete 5ms-frame forwarded
start operation
whole-band interference decision
wait for 'start' signal
State_i0
initialization of the internal control signals and counters
first peak value
interference decision finalized
State_i1
1) number_of_peaks=+1;
2) If current peak > predefined interference-threshold then number_of_valid_peaks=+1;
End if;
number_of_peaks < 5ms-frame_length
whole 5ms-frame processed
State_i2
wait for one OFDM symbol (i.e., 2560 clock cycles)
State_i3
If number_of_valid_peaks < 5ms-frame_length then interference detected=1;
End if;
Wholeband interference-detection

Figure 6.13: Indicative diagram of the designed control state machines.

6.4 Integration and implementation using the GEDOMIS testbed

The integration of the baseband design to the firmware of the GEDOMIS' ADP boards was conducted in a similar way to the one described in Section 5.4. The main difference resides in the precise partition of the FPGA implementation and the exact testbed configuration utilized to deploy the presented macrocell/femtocell interference-mitigation scheme.

6.4.1 Real-time channel and interference emulation

The EB Propsim C8 channel emulator plays a key role in the reproduction of the considered channel propagation and interference conditions. In order to recreate the specific interference scenario, the following configuration has been utilized:

(i) An ITU Pedestrian B channel model was applied to the RF signal generated by the macro BS emulating both static and mobile channel conditions.

(ii) A custom channel model is applied to the femto DL signal in order to produce one of the four possible interference scenarios. In more detail, two channel models were defined, one with a response similar to a high-pass filter and another with a response similar to a low-pass filter; keeping one of the two 10 MHz bands of the femto-generated signal. Hence, a static channel impulse response is considered.

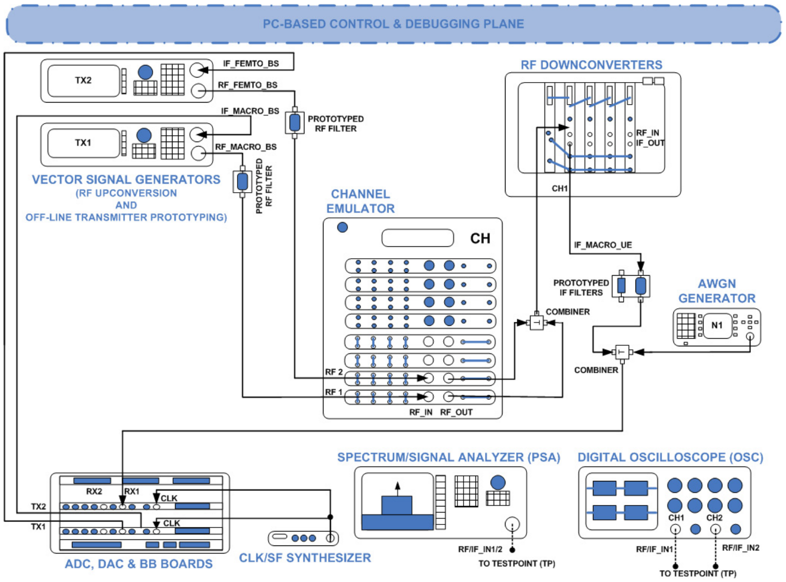


Figure 6.14: Specific GEDOMIS® testbed setup utilized to recreate the macrocell/femtocell LTE-based scenario.

(iii) The two corresponding RF outputs of the channel emulator are finally combined to provide the DL signal that is received by the macro UE.

The setup of GEDOMIS representing the previously described configuration is shown in Figure 6.14. Additionally, the channel emulator can be programmed at run-time to provide a precise SIR level, by configuring the output gain of its internal RF upconversion stage (i.e., the user is able to adjust the input/output gains of the channel emulator). Moreover, the gain level of different amplifiers and attenuators encountered in the entire signal path of the testbed were exhaustively tested for every 1 dB attenuation step within the previously defined SIR range (i.e., between 12 and 20 dB). Finally, the precise time-shift (i.e., in samples) of the DL macro and interference signals is fully controlled using a user-programmable register which is located in the baseband FPGA implementation of the two BSs.

6.4.2 FPGA-based implementation of the macrocell/femtocell interference-mitigation scheme

Figure 6.15 illustrates the multi-FPGA partitioning of the implemented PHY-layer prototype, which hosts the LTE-based macrocell/femtocell interference-mitigation scheme. As it can be observed, both the macro and femto BSs are implemented in the FPGA device of the VHS-DAC board, whereas the macro UE utilizes two programmable devices. A fourth FPGA is utilized to facilitate the high-throughput data-forwarding between the partitions of the system residing in the VHS-ADC and SMQUAD boards and also to implement the emulated UL communication (i.e., dedicated feedback link between the macro UE and femto BS). The FPGA-resource utilization of the four FPGA devices is detailed in Table 6.2.

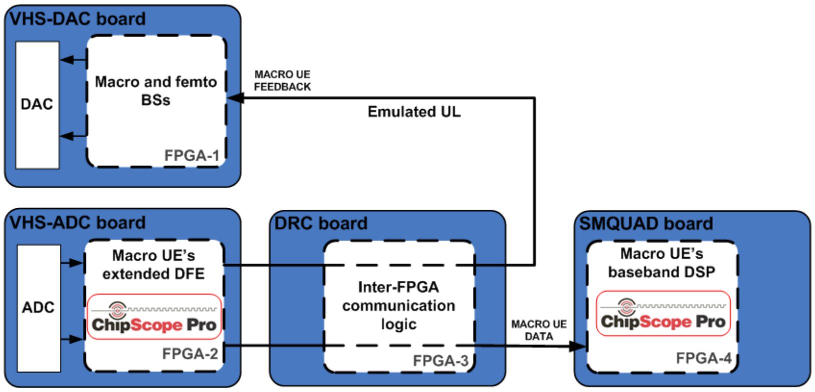


Figure 6.15: Multi-FPGA partitioning of the implemented LTE-based macrocell/femtocell interference-mitigation scheme.

	Macro/Femto BSs	Macro UE		
	FPGA 1	FPGA 2	FPGA 3	FPGA 4
	XC4VLX160	*XC4VLX160*	*XC4VSX35*	*XC4VLX160*
Slices	27%	75%	20%	33%
DSP48s	82%	48%	1%	94%
RAMB16s	78%	78%	55%	62%

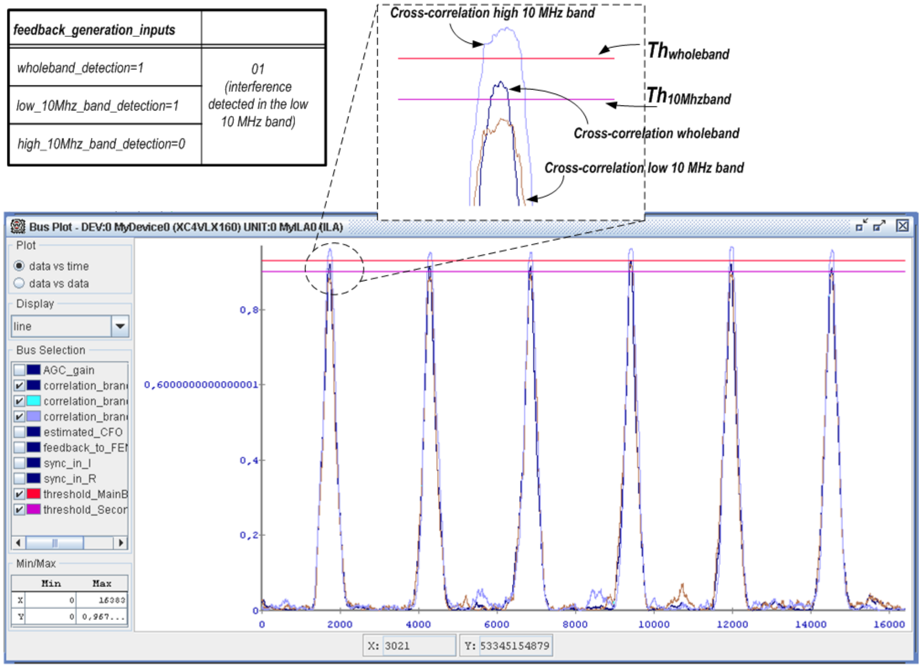
Table 6.2: FPGA utilization indicators of the implemented system.

6.5 Experimental results

As it was described before, the channel emulator was configured to provide a channel response between the macro BS and the macro UE according to the ITU Pedestrian B model. Furthermore, different interference scenarios have also been tested for both static and low-mobility conditions.

A number of ChipScope FPGA-monitoring cores, embedded within the baseband design, have been used to get an insight of the interference-detection operation. Figure 6.16 is a screen capture of the Chipscope software which shows the evolution of the cross-correlation values in a scenario where interference is forced in the low 10 MHz band, with a SIR of 12 dB, while applying a static Pedestrian B channel model for the macro DL communication. As it can be observed, the correlation peaks are below the defined interference-detection thresholds, hence indicating that the DL signal of the macro UE is being interfered by the femto communication.

During the experimental verification of the system it has been also utilized a real-time digital oscilloscope to inspect the generated RF signals (both at time and frequency domain). Figure 6.17 an indicative screen capture extracted from the mentioned instrument that demonstrates how the PRBs of the femto DL signal are adapted in real-time, when interference is detected (i.e., as already described before, the transmitter is notified about the occurrence of this event through the feedback mechanism). The monitoring of the channel and interference effects was also facilitated by using a RF spectrum analyzer, as it is shown in the respective screen capture in Figure 6.18; more specifically, the blue line



feedback_generation_inputs	
wholeband_detection=1	01 (interference detected in the low 10 MHz band)
low_10Mhz_band_detection=1	
high_10Mhz_band_detection=0	

Bus Plot - DEV:0 MyDevice0 (XC4VLX160) UNIT:0 MyILA0 (ILA)

Plot
● data vs time
○ data vs data

Display
line

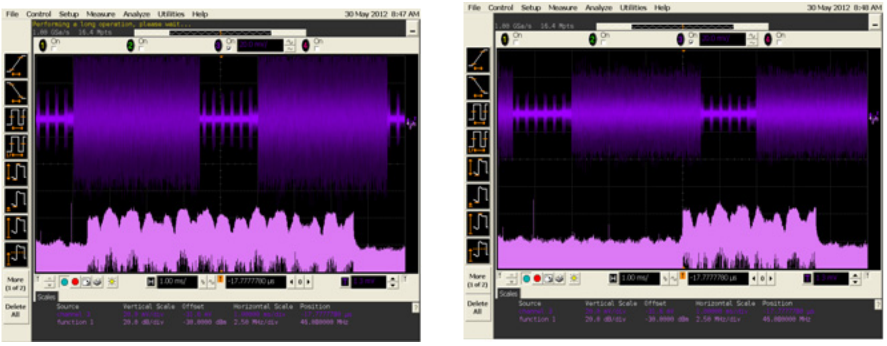
Bus Selection
□ AGC_gain
☑ correlation_bran
☑ correlation_bran
☑ correlation_bran
□ estimated_CFO
□ feedback_to_FE
□ sync_in_I
□ sync_in_R
☑ threshold_MainB
☑ threshold_Secor

Min/Max

	Min	Max
X	0	16383
Y	0	0,967...

X: 3021 Y: 53345154879

Figure 6.16: Cross-correlation versus the defined interference-detection thresholds for the 3 correlation chains at the macro UE.



(a) Whole-band transmission.

(b) The high 10 MHz band is used.

Figure 6.17: Oscilloscope capture of the interference-aware femto DL signal.

represents the RF spectrum of the macro DL signal under ideal signal propagation conditions, the black line represents the same signal after a mobile channel (i.e., 3 km/h) has been applied and, finally, the green line shows the degraded spectrum in the presence of interference in the low 10 MHz band under mobile channel conditions.

In order to facilitate the experimental analysis of the proposed scheme, two transmission-modes were defined for the femto BS; one is based on the received feedback by the macro UE (i.e., applying adaptive PRB allocation) and the other forces a whole-band transmission (i.e., the feedback is ignored). Hence, during the experimental validation, the femto DL signal was generated by one of

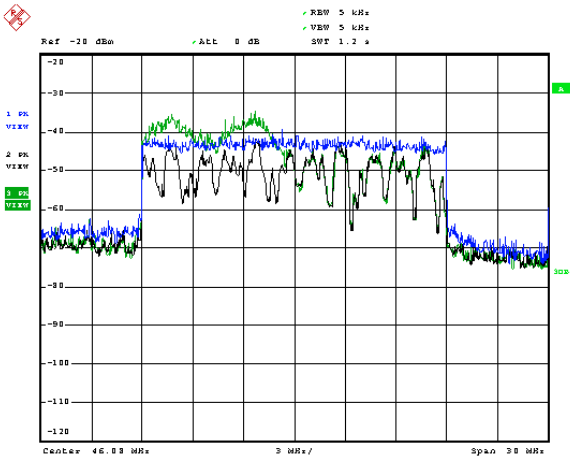
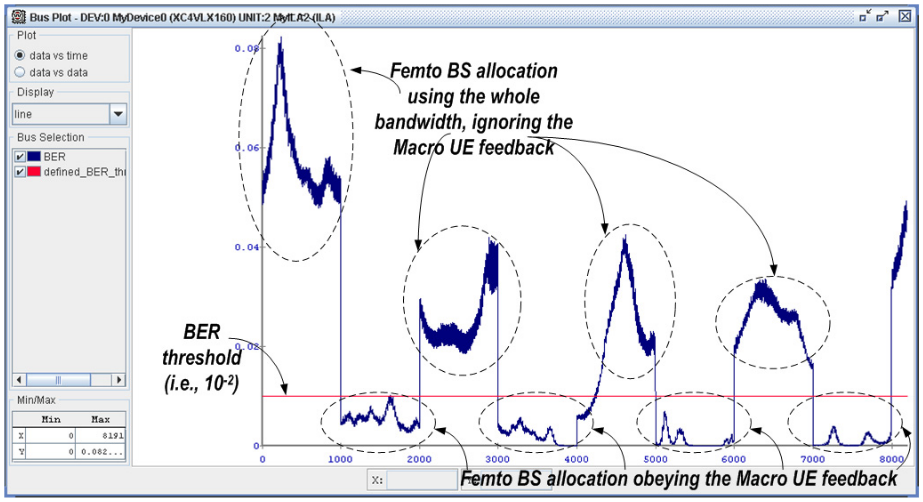


Figure 6.18: Visualization of the impairments introduced to the macro DL signal by both the mobile channel and the interference signal.

the two transmission-modes in fixed-length time-periods (i.e., this time-period could be modified on-the-fly through a user-controlled programmable register).

Considering the need to evaluate the proposed test scenario under mobile channel conditions, and in order to avoid an extensive measurement campaign (such as the one featured in the mobile WiMAX system), it has been decided to add additional logic in the macro UE to calculate the instantaneous BER (in a per frame basis); in order to achieve this, the PRBS generator of the macro BS was replicated in the receiver and its output was compared with the de-mapper one. This real-time calculation of the BER metric avoids the tedious and lengthy data-capturing and post-processing; it also provided a great instrument to test, debug and validate the proposed interference-management scheme at run-time (i.e., by utilizing the ChipScope Pro software tool to visualize it). Figure 6.19 shows a representative Chipscope screen capture of the observed BER at the macro UE under the same interference conditions described before, for a low-mobility (i.e., 0.2 km/h) realization of the considered Pedestrian B channel model. The figure covers a period of 40 seconds (i.e., 8000 5 ms-frames). It has to be noted that the transmission-mode of the femto BS changes each 5 seconds. Moreover, it can be observed how the KPI described in Section 6.3.2 is fulfilled during the periods where the PRB allocation of the femto DL signal is adapted according to the received macro UE feedback.

Similarly, in Figure 6.20 it is repeated the experiment for a static channel realization, where a SIR of 10 dB is applied considering an interference on the whole 20 MHz band. Moreover, the evolution of the observed BER level for different SIR values (i.e., from 10 to 14 dB), in static channel conditions is provided by Figures 6.21a, 6.21b and 6.21c, where the considered interference is taking place in the upper 10 MHz band. Finally, a mobile channel (i.e., 3 km/h) and a SIR of 14 dB is applied to the same scenario, as it can be observed in Figure 6.21d.



Bus Plot - DEV:0 MyDevice0 (XC4VLX160) UNIT:2 MyILA2 (ILA)

Plot
- ● data vs time
- ○ data vs data

Display: line

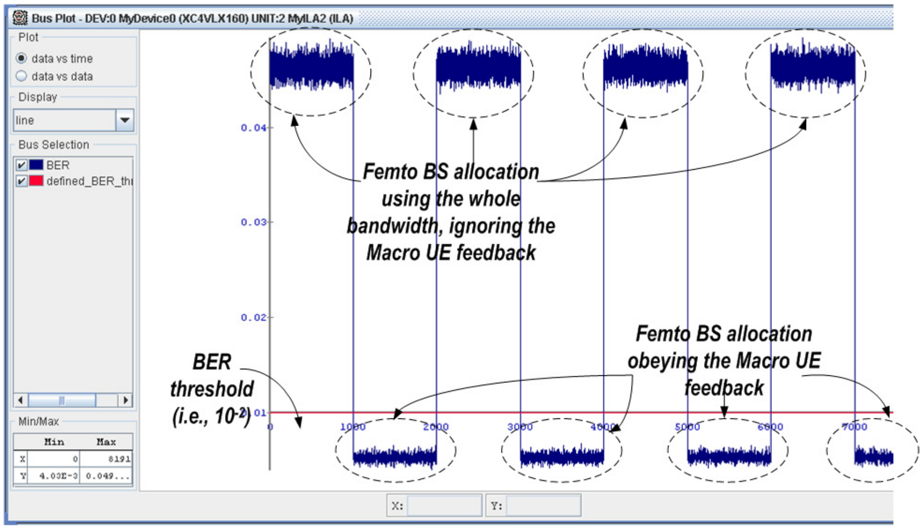
Bus Selection
- ☑ BER
- ☑ defined_BER_thr

Min/Max

	Min	Max
X	0	8191
Y	0	0.082...

0.08
0.06
0.04
0.02
0
0
1000
2000
3000
4000
5000
6000
7000
8000
Femto BS allocation using the whole bandwidth, ignoring the Macro UE feedback
BER threshold (i.e., 10^{-2})
Femto BS allocation obeying the Macro UE feedback
X:

Figure 6.19: ChipScope screen capture depicting the macro UE BER under low mobility conditions and a SIR of 12 dB.



Bus Plot - DEV:0 MyDevice0 (XC4VLX160) UNIT:2 MyILA2 (ILA)

Plot
- ● data vs time
- ○ data vs data

Display: line

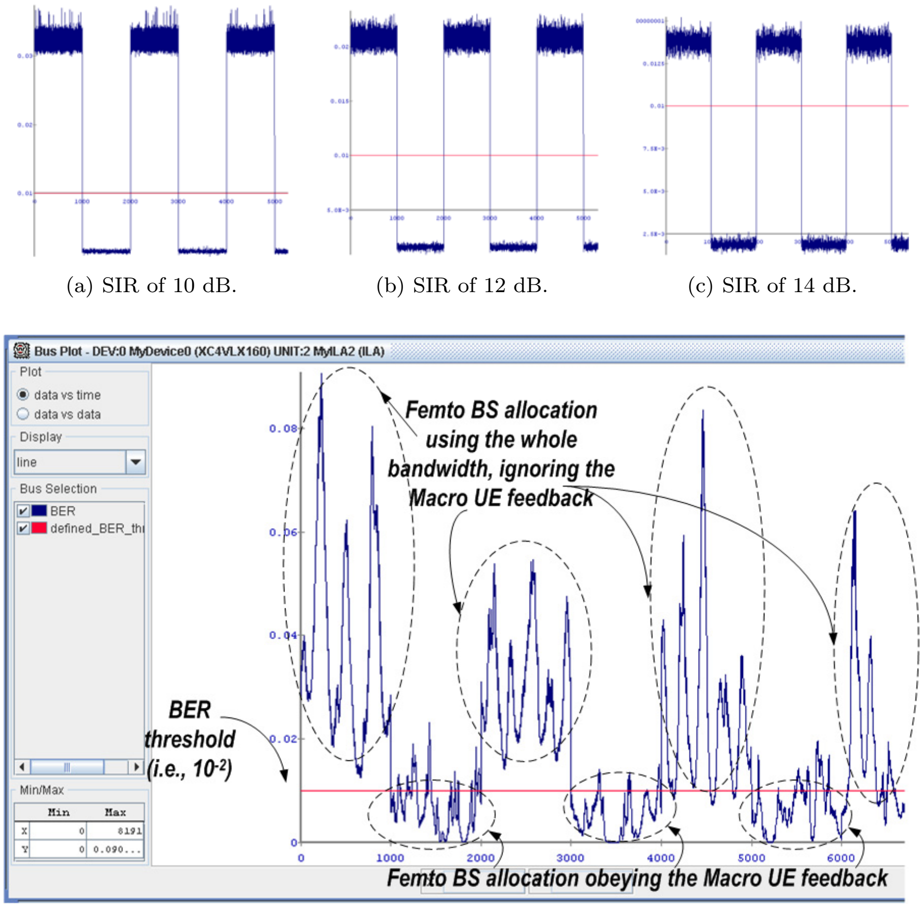
Bus Selection
- ☑ BER
- ☑ defined_BER_thr

Min/Max

	Min	Max
X	0	8191
Y	4.03E-3	0.049...

0.04
0.03
0.02
0.01
0
1000
2000
3000
4000
5000
6000
7000
Femto BS allocation using the whole bandwidth, ignoring the Macro UE feedback
BER threshold (i.e., 10^{-2})
Femto BS allocation obeying the Macro UE feedback
X:
Y:

Figure 6.20: ChipScope screen capture depicting the macro UE BER under static channel conditions and a SIR of 10 dB.



Bus Plot - DEV:0 MyDevice0 (XC4VLX160) UNIT:2 MyILA2 (ILA)

● data vs time
○ data vs data

	Min	Max
X	0	8191
Y	0	0.090...

(a) SIR of 10 dB.

(b) SIR of 12 dB.

(c) SIR of 14 dB.

(d) Mobile conditions and SIR of 14 dB.

Figure 6.21: ChipScope screen capture depicting the macro UE BER for different SIR values and mobility conditions.

Chapter 7

Conclusions and future work lines

This chapter compiles the main conclusions of the innovating solutions introduced in this thesis. The goal of these solutions was to maximize the parallelization and resource re-utilization of a number of DSP algorithms that form part of certain PHY-layer building blocks, in order to enable the efficient baseband design of modern wireless communication systems. Finally, some possible future work lines are proposed to extend the reach of the presented research.

7.1 Conclusions

The different approaches commonly utilized to design, implement and validate the PHY-layer of indicative modern wireless communication systems were introduced in Chapter 2. While each approach could result more convenient for the objectives of specific use cases, the common ground that such systems share is that, starting from their theoretical conception and arriving up to their physical implementation and real-time validation, different degrees, abstractions or levels of innovation are required. Specifically, in this thesis the focus was laid on the real-time prototyping of high-performance PHY-layer schemes requiring run-time adaptivity (according to the instantaneous channel conditions). Given their inherent parallelism and flexibility, FPGAs were selected as the target technology. Thus, the principal motivation of this thesis was the provision of innovation at the digital design level to provide an efficient realization of the considered systems (i.e., based on the MIMO-OFDM technology). In this context efficiency was translated to intelligent architectural and RTL design decisions that resulted in notable savings in FPGA area, embedded memory blocks and dedicated DSP48 slices. To achieve this, several critical design novelties were applied, including resource sharing strategies, concurrent re-utilization of various baseband processing blocks and parallelization or simplification of various arithmetic operations among others. Furthermore, the utilization of realistic signal propagation conditions and hardware constraints has played a fundamental role in the validation of the developed use-cases at system-level, since the operating and testing conditions were configured to be as close as possible to the ones encountered in real-life application scenarios; in fact, these conditions had a direct impact on the design, implementation and validation cycle adding degrees of freedom to the already complex baseband development.

Chapter 3 has provided a taxonomy of the related literature. The review of relevant research has covered all the steps from the high-level modelling to the ASIC implementation of advanced PHY-layer algorithms and systems. Innovative solutions were encountered for each of these isolated steps, where the different utilized simulation and implementation technologies (i.e., PHY-layer development ecosystem) is presenting its own limitations and difficulties. Moreover, different assumptions or simplifications are considered by the different approaches utilized at each step; for this reason, it was important to compile a detailed list of the objectives, assumptions and conditions that were utilized in the developed case studies, in order to underline the ways that this thesis differs from the related work of other researchers. The contributions of this thesis have also been clearly marked up, especially for those related to the RTL-design techniques that were employed to enable a performance-efficient FPGA implementation. Similarly, the critical role of the design, implementation and verification methodology was also highlighted and described in full detail. The overall contribution of this thesis has been graphically summarized, in relation to the reviewed research initiatives.

A very important companion that enabled the innovating low-level digital baseband design was the proposed methodology (Chapter 4) which is tailored for a well-structured, iterative, incremental and modular development flow, covering exhaustive and precise testing and validation at all of its stages. The proposed development flow benefits from the combination of different abstraction-level approaches, allowing by this way to address a number of the previously identified challenges and limitations. In fact, the proposed and employed methodology plays a key role in the successful development and validation of the high-performance systems presented in the two use-case chapters. Another key point of this methodology, is that it is directly transferable and reusable in similar custom HDL developments of broadband PHY-layer systems that feature increased computational complexity, high degree of run-time in-system adaptivity and dense FPGA implementation requirements.

The work presented herein has critically contributed in the efficient baseband design of modern wireless communication systems. This was achieved by simplifying and optimizing a number of arithmetic calculation featured in DSP algorithms. The latter form part of certain baseband building blocks, which are encountered in modern OFDM-based communication systems. This not only allowed to meet the stringent real-time performance requirements, but also enabled the intelligent re-utilization, resource sharing or cost-effective parallelization of the processing resources and memory components available at the target FPGA devices.

In more details, the incremental development of a 2x2 MIMO closed-loop PHY-layer scheme, based on the mobile WiMAX wireless communication standard, provided the first use case of the proposed development methodology. Moreover, a second use case is provided by implementing an LTE-based Macrocell/Femtocell interference-mitigation scheme. This second use case was built upon a design and IP re-utilization incremental development strategy, that took advantage of the outcome of the first use case. A series of architectural solutions have been employed towards a performance-efficient FPGA-based implementation of the target baseband systems, as detailed in the following:

(i) Significant resource-sharing, signal processing parallelization and FPGA area-savings were achieved by optimizing the implementation of selected algorithms or arithmetic operations at RTL-level. In other words, a streamlined formulation was applied, leading to a reduced consumption of computational resources (especially of embedded DSP-slices) without compromising the overall performance of the system.

(ii) To fully leverage the benefits of computational complexity-bounded algorithms, custom processing architectures were defined and optimized at low-level(e.g., efficiently use FPGA-resources to enable a high-speed pipelined computation and/or seek a minimized execution latency). In order to achieve this, the custom RTL description was complemented with gate-level design when the stringent processing or low-latency requirements were imposing a fully optimized implementation.

(iii) The utilization of large bandwidths resulted in demanding baseband communications, which required the definition of high-throughput data-paths utilizing complex memory structures. The developed solution to deal with this challenge, was an intelligent utilization of adaptive memory structures, optimizing likewise the utilization of the embedded RAM blocks while facilitating the interaction between the DSP stages. The memory-accesses were extended with additional DSP-functionalities by embedding dedicated control logic to the defined memory structures. In a similar way, the definition of in-block dedicated memory structures played a crucial role in the reduction of the overall latency of the designed pipelined DSP processing stages (e.g., by providing the required operands without extra latencies, accounting for the data interdependencies).

(iv) Given the substantial computational capacity required at baseband, and the size of the target FPGA devices, it was essential to provide an optimized utilization of the available FPGA-resources. This was achieved by identifying first those DSP operations that were able to benefit by applying time-multiplexing of their underlying computational resources (e.g., applying arithmetic optimizations to certain baseband algorithms or re-utilizing processing blocks that are shared among all MIMO-OFDM baseband systems). Furthermore, they were exploited similarities at processing-block level of DSP operations which were utilized by different communication schemes.

(v) Apart from the large amount of concurrent synchronous operations that were executed at the developed baseband systems, it was also efficiently addressed the dynamic adaptation of the underlying DSP logic (e.g., change between communication schemes accounting for the current channel conditions). Consequently, it was implemented a centralized control unit to manage the operation of the diverse DSP stages and their synchronous communications. The centralized controller accounted for the internal processing latencies and the formatting of each of the utilized frame configurations. Another important function of the controller was to manage the transmission/reception of the feedback information utilized in the closed-loop communication schemes. A hierarchy of dedicated state machines and status registers allowed its efficient implementation.

An important feature of the proposed RTL design techniques is that they are not confined to the implementation of wireless communication systems, but can be extrapolated to other use cases where bit-intensive DSP needs to be employed. Finally, it must be underlined how the presented use cases provide proof-of-concept of advanced PHY-layer techniques which are meant to conform a vital part of the baseband of future wireless communication systems (e.g., multi-antenna communications, adaptation of the DSP in response to the perceived channel conditions or interference-mitigation in a heterogeneous wireless access-technology scenario).

7.2 Future work lines

The implementation of the PHY-layer of modern and future wireless communication systems can benefit from the provided RTL design techniques to attain the required hardware efficiency. Furthermore, the quality of the resulting development can be extended by considering the presented incremental design, implementation and verification methodology. Yet, further advances are required at digital design level considering the future evolution of wireless communications. Hence, some interesting future work lines have been identified to extend the research conducted in this thesis, as detailed below:

- The developed baseband systems can be extended to include more advanced DSP schemes and functionalities (e.g., AMC or SM). Eventually, multi-user scenarios are to be targeted to realistically analyse the benefits of advanced MIMO closed-loop schemes including cognitive functionalities.

- Adding more cognitive functionalities considering either point-to-multipoint or multipoint-to-multipoint communication links would require substantial enforcement and top-up flexibility for the centralized control unit. This naturally would require the inclusion of a real-time MAC-layer, whose development could be based on the design and implementation methodology introduced in this thesis (e.g., towards a HW/SW co-design approach).

- Apart from accounting the perceived channel conditions to trigger the dynamic adaptation of the DSP operations, other aspects may also be considered. Indicatively, a centralized energy-aware controller can be designed to reduce the overall energy consumption of the UE. The latter can be achieved by considering, on top of the CSI, a number of cross-layer inputs, in order to select the optimum PHY-layer scheme to be utilized at each time instant. Indicatively, these cross-layer inputs may include network operator parameters and user requests (i.e., QoS levels), performance metrics, information regarding the communications environment (e.g., interference profile) and parameters describing the energy status of the UE (e.g., current and past battery status, energy consumption rate).

Bibliography

Reference publications

[1] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang, "Incentivizing Time-Shifting of Data: A Survey of Time-Dependent Pricing for Internet Access," *IEEE Communications Magazine*, vol. 40, no. 11, pp. 91–99, Nov. 2012.

[2] Z. Abichar, J. M. Chang, and C.-Y. Hsu, "WiMAX or LTE: Who will Lead the Broadband Mobile Internet?" *IT Professional*, vol. 12, no. 3, pp. 26–32, Jun. 2010.

[3] P. L. Gilabert, G. Montoro, D. López, N. Bartzoudis, E. Bertrand, M. Payaró, and A. Hourtane, "Order Reduction of Wideband Digital Predistorters Using Principal Component Analysis," in *Proc. IEEE International Microwave Symposium (IMS)*, Jun. 2013.

[4] V. Jungnickel, A. Forck, T. Haustein, S. Schiffermüller, C. von Helmolt, F. Luhn, M. Pollock, C. Juchems, M. Lampe, S. Eichinger, W. Zirwas, and E. Schulz, "1 Gbit/s MIMO-OFDM Transmission Experiments," in *Proc. IEEE Vehicular Technology Conference (VTC)*, Sep. 2005.

[5] J. Inkeles, "Implementing FPGA Design with the OpenCL Standard," White paper, Altera®, November 2012.

[6] S. McGettrick, K. Patel, and C. Bleakley, "High Performance Programmable FPGA overlay for Digital Signal Processing," in *Proc. International Conference on Applied Reconfigurable Computing: Architectures, Tools and Applications (ARC)*, Mar. 2011.

[7] K. DeHaven, "EPPs: The Ideal Solution for a Wide Range of Embedded Systems," White paper, Xilinx, June 2012.

[8] F. Pratas, A. Iliĉ, L. Sousa, and H. C. Neto, "Double-precision Floating-point Performance of Computational Devices: FPGAs, CPUs, and GPUs," in *Proc. Jornadas sobre Sistemas Reconfiguráveis (REC)*, Feb. 2010.

[9] P. G. Jr., R. Hang, D. Truhachev, and C. Schlegel, "A Portable MIMO Testbed and Selected Channel Measurements," *EURASIP Journal on Applied Signal Processing*, vol. 2006, 2006.

[10] C. Mehlführer, J. Colom Ikuno, M. Ŝimko, S. Schwarz, M. Wrulich, and M. Rupp, "The Vienna LTE simulators - Enabling reproducibility in wireless communications research," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, 2011.

[11] C. Mehlführer, S. Caban, and M. Rupp, “Experimental Evaluation of Adaptive Modulation and Coding in MIMO WiMAX with Limited Feedback,” *EURASIP Journal on Applied Signal Processing*, vol. 2008, 2008.

[12] M. Parker, “Taking Advantage of Advances in FPGA Floating-Point IP Cores,” White paper, Altera®, October 2009.

[13] T. Vanevenhoven, “High-Level Implementation of Bit- and Cycle-Accurate Floating-Point DSP Algorithms with Xilinx FPGAs,” White paper, Xilinx, October 2011.

[14] C. Moy and M. Raulet, “High-Level Design Methodology for Ultra-Fast Software Defined Radio Protoyping on Heterogeneous Platforms,” *Journal on Advances in Electronics and Telecommunications*, vol. 1, no. 1, pp. 67–85, Apr. 2010.

[15] E. Mohsen, “Scaling High-Performance Applications for Low Power and Cost,” White paper, Xilinx, July 2012.

[16] J. Hussein, M. Klein, and M. Hart, “Lowering Power at 28nm with Xilinx 7 Series FPGAs,” White paper, Xilinx, February 2012.

[17] X. Chu and J. McAllister, “FPGA Based Soft-Core SIMD Processing: A MIMO-OFDM Fixed-Complexity Sphere Decoder Case Study,” in *Proc. International Conference on Field Programmable Technology (FPT)*, Dec. 2010.

[18] H. Y. Cheah, S. A. Fahmy, and D. L. Masell, “iDEA: A DSP Block Based FPGA Soft Processor,” in *Proc. International Conference on Field Programmable Technology (FPT)*, Dec. 2012.

[19] A. George, H. Lam, and G. Stitt, “Novo-G: At the Forefront of Scalable Reconfigurable Supercomputing,” *Computing in Science & Engineering*, vol. 13, no. 1, pp. 82–86, Feb. 2011.

[20] J. K. Lee and G. D. Peterson, “Iterative Refinement on FPGAs,” in *Proc. Symposium on Application Accelerators in High-Performance Computing (SAAHPC)*, Jul. 2011.

[21] J. Flowers, G. Brown, P. Cooke, and G. Stitt, “A performance and energy comparison of FPGAs, GPUs, and multicores for sliding-window applications,” in *Proc. ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA)*, Feb. 2012.

[22] M. Rupp, S. Caban, and C. Mehlführer, “Challenges in Building MIMO Testbeds,” in *Proc. European Signal Processing Conference (EUSIPCO)*, Sep. 2007.

[23] A. Sghaier, S. Areibi, and R. Dony, “Implementation Approaches Trade-Offs for WiMax OFDM Functions on Reconfigurable Platforms,” *ACM Transactions on Reconfigurable Technology and Systems*, vol. 3, no. 3, Sep. 2010.

[24] A. Engel, B. Liebig, and A. Koch, "Feasibility Analysis of Reconfigurable Computing in Low-Power Wireless Sensor Applications," in *Proc. International Symposium on Applied Reconfigurable Computing (ARC)*, Mar. 2011.

[25] H. Gadke and A. Koch, "Comrade - A Compiler for Adaptive Computing Systems using a Novel Fast Speculation Technique," in *Proc. International Conference on Field Programmable Logic and Applications (FPL)*, Aug. 2007.

[26] J. Noguera, S. Neuendorffer, S. V. Haastregt, J. Barba, K. Vissers, and C. Dick, "Implementation of Sphere Decoder for MIMO-OFDM on FPGAs Using High-level Synthesis Tools," *Analog Integrated Circuits and Signal Processing*, vol. 69, no. 3, pp. 119–129, Sep. 2011.

[27] M. Fernandez and P. Abusaidi, "Virtex-6 FPGA Routing Optimization Design Techniques," White paper, Xilinx, October 2010.

[28] V. Elvira, J. Ibáñez, I. Santamaría, K. Tittelbach-Helmrich, and Z. Stamenkovic, "Baseband Processor for RF-MIMO WLAN," in *Proc. IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, Dec. 2010.

[29] O. Fernández, M. Domingo, and R. P. Torres, "Outdoor to indoor 2x2 Wideband MIMO Channel Modelling," in *Proc. IEEE Vehicular Technology Conference (VTC)*, Sep. 2009.

[30] P. Murphy, A. Sabharwal, and B. Aazhang, "Design of WARP: A Wireless Open-Access Research Platform," in *Proc. European Signal Processing Conference (EUSIPCO)*, Sep. 2006.

[31] A. Khattab, J. Camp, C. Hunter, P. Murphy, A. Sabharwal, and E. W. Knightly, "WARP - A Flexible Platform for Clean-Slate Wireless Medium Access Protocol Design," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 12, no. 1, pp. 56–58, Jan. 2008.

[32] J. A. García-Naya, O. Fresnedo, and F. J. Vazquez-Araujo, "Experimental Evaluation of Analog Joint Source-Channel COding in Indoor Environments," in *Proc. IEEE International Conference on Communications (ICC)*, Jun. 2011.

[33] M. Wenk, P. Luethi, T. Koch, P. Maechler, N. Felber, W. Fichtner, and M. Lerjen, "Hardware Platform and Implementation of a Real-Time Multi-User MIMO-OFDM Testbed," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2009.

[34] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. F. and: S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated Multipoint: Concepts, Performance, and Field Trial Results," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 102–111, Feb. 2011.

[35] V. Jungnickel, L. Thiele, T. Wirth, T. Haustein, , A. Forck, S. Wahls, S. Jaeckel, S. Schubert, H. Gäbler, C. Juchems, F. Luhn, R. Zavrtak,

H. Droste, G. Kadel, W. Kreher, J. Mueller, W. Stoermer, and G. Wannemacher, "Coordinated Multipoint Trials in the Downlink," in *Proc. IEEE Broadband Wireless Access Workshops (BWAWS), co-located with IEEE GLOBECOM*, Nov. 2009.

[36] T. Wirth, V. Jungnickel, A. Forck, S. Wahls, V. Venkatkumar, T. Haustein, and H. Wu, "Polarisation Dependent MIMO Gains on Multiuser Downlink OFDMA with a 3GPP LTE Air Interface in Typical Urban Outdoor Scenarios," in *Proc. International ITG Workshop Smart Antennas (WSA)*, Feb. 2008.

[37] E. Aschbacher, M. Y. Cheong, P. Brunmayr, M. Rupp, and T. I. Laakso, "Prototype Implementation of Two Efficient Low-Complexity Digital Predistortion Algorithms," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008.

[38] S. Caban, C. Mehlführer, R. Langwieser, A. L. Scholtz, and M. Rupp, "Vienna MIMO Testbed," *EURASIP Journal on Applied Signal Processing*, vol. 2006, 2006.

[39] S. Caban, C. Mehlführer, G. Lechner, and M. Rupp, "Testbedding MIMO HSDPA and WiMAX," in *Proc. IEEE Vehicular Technology Conference (VTC)*, Sep. 2009.

[40] N.-U.-I. Muhammad, R. Rasheed, R. Pacalet, R. Knopp, and K. Khalfallah, "Flexible Baseband Architectures for Future Wireless Systems," in *Proc. Euromicro Conference on Digital System Design (DSD)*, Sep. 2008.

[41] B. B. Romdhanne, N. Nikaein, R. Knopp, and C. Bonnet, "OpenAirInterface Large-Scale Wireless Emulation Platform and Methodology," in *Proc. ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks (PM2HW2N)*, Oct. 2011.

[42] W. Gabran and B. Daneshrad, "Hardware and Physical Layer Adaptation for a Power Constrained MIMO OFDM System," in *Proc. IEEE International Conference on Communications (ICC)*, Jun. 2011.

[43] J. Reitterer and M. Rupp, "Interference Alignment in UMTS Long Term Evolution," in *Proc. European Signal Processing Conference (EUSIPCO)*, Aug. 2011.

[44] M. Ŝimko, Q. Wang, and M. Rupp, "Optimal Pilot Symbol Power Allocation under Time-Variant Channels," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, 2012.

[45] J. C. Ikuno, S. Pendl, M. Ŝimko, and M. Rupp, "Accurate SINR Estimation Model for System Level Simulation of LTE Networks," in *Proc. IEEE International Conference on Communications (ICC)*, Jun. 2012.

[46] C. Hunter and A. Sabharwal, "Distributed Protocols for Interference Management in Cooperative Networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 9, pp. 1633–1640, Oct. 2012.

[47] B. Zayen, B. Kouassi, R. Knopp, F. Kaltenberger, D. Slock, I. Ghauri, and L. Deneire, "Software Implementation of Spatial Interweave Cognitive Radio Communication using OpenAirInterface Platform," in *Proc. International Symposium on Wireless Communication Systems (ISWCS)*, Aug. 2012.

[48] J. P. González-Coma, P. M. Castro, and L. Castedo, "Transmit Impairments Influence on the Performance of MIMO Receivers and Precoders," in *Proc. European Wireless Conference (EW)*, Apr. 2011.

[49] P. Zetterberg, "Experimental Investigation of TDD Reciprocity-Based Zero-Forcing Transmit Precoding," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, 2011.

[50] S. Hu, G. Wu, Y. L. Guan, C. L. Law, Y. Yan, and S. Li, "Development and Performance Evaluation of Mobile WiMAX Testbed," in *Proc. IEEE Mobile WiMAX Symposium*, Mar. 2007.

[51] M. E. Şahin and H. Arslan, "MIMO-OFDMA Measurements; Reception, Testing, and Evaluation of WiMAX MIMO Signals With a Single Channel Receiver," *IEEE Trans. on Instrumentation and Measurement*, vol. 58, no. 3, pp. 713–721, Mar. 2009.

[52] G. Fettweis, J. Holfeld, V. Kotzsch, P. Marsch, E. Ohlmer, Z. Rong, and P. Rost, "Field Trial Results for LTE-Advanced Concepts," in *Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Mar. 2010.

[53] M. Duarte, P. Murphy, C. Hunter, S. Gupta, and A. Sabharwal, "WARPLab: Multi-Node Prototyping with Real Wireless Data," *ELSEVIER Journal of the Franklin Institute (submitted)*, Oct. 2011.

[54] M. Duarte, C. Dick, and A. Sabharwal, "Experiment-Driven Characterization of Full-Duplex Wireless Systems," *IEEE Trans. on Wireless Communications (submitted)*, May 2012.

[55] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly, "Design and Experimental Evaluation of Multi-User Beamforming in Wireless LANs," in *Proc. ACM International Conference on Mobile Computing and Networking (MobiCom)*, Sep. 2010.

[56] J. Luo, A. Kortke, W. Keusgen, J. Li, M. Haard, P. Prochazka, and J. Skyora, "A Flexible Hardware-In-the-Loop Test Platform for Physical Resource Sharing Mechanisms in Wireless Networks," in *Proc. Future Network & Mobile Summit (FutureNetw)*, Jun. 2011.

[57] C. Studer, M. Wenk, and A. Burg, "MIMO Transmission with Residual Transmit-RF Impairments," in *Proc. International ITG Workshop on Smart Antennas (WSA)*, Feb. 2010.

[58] Q. Wang, S. Caban, C. Mehlführer, and M. Rupp, "Measurement Based Throughput Evaluation of Residual Frequency Offset Compensation in WiMAX," in *Proc. International Symposium of the Croatian Society Electronics in Marine (ELMAR)*, Sep. 2009.

[59] C. Mehlführer, S. Caban, and M. Rupp, "Measurement-Based Performance Evaluation of MIMO HSDPA," *IEEE Trans. on Vehicular Technology*, vol. 59, no. 9, pp. 4354–4367, Nov. 2010.

[60] S. Caban, J. A. García-Naya, C. Mehlführer, L. Castedo, and M. Rupp, "A Real-Time FPGA-Based Implementation of a High-Performance MIMO-OFDM Mobile WiMAX Transmitter," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT), Revised Selected Papers*, vol. 45, pp. 388–399, May 2010.

[61] C. Mehlführer, S. Caban, J. A. García-Naya, and M. Rupp, "Throughput and Capacity of MIMO WiMAX," in *Proc. Asilomar Conference on Signals, Systems and Computers (ACSSC)*, Nov. 2009.

[62] F. J. Vázquez-Araujo, J. A. García-Naya, M. González-López, L. Castedo, and J. Garcia-Frias, "Experimental Evaluation of MIMO Coded Modulation Systems: Concatenation with OSTBC or Spatial Multiplexing," in *Proc. International Conference on Systems, Signals and Image Processing (IWSSIP)*, Apr. 2012.

[63] J. Gutiérrez, O. González, J. Pérez, D. Ramírez, L. Vielva, J. Ibáñez, and I. Santamaría, "Frequency-Domain Methodology for Measuring MIMO Channels Using a Generic Test Bed," *IEEE Trans. on Instrumentation and Measurement*, vol. 60, no. 3, pp. 827–838, Mar. 2011.

[64] H. J. Pérez-Iglesias, J. A. García-Naya, A. Dapena, and J. C. Brégains, "Data Transmission over a Wireless MIMO Testbed with Alamouti Code: Pefromance Comparison of Channel Estimation Algorithms," in *Proc. Workshop on Multimedia Data Coding and Transmission (WMDCT)*, Sep. 2010.

[65] J. A. García-Naya, L. Castedo, O. González, D. Ramírez, and I. Santamaria, "Experimental Evaluation of Interference Alignment under Imperfect Channel State Information," in *Proc. European Signal Processing Conference (EUSIPCO)*, Aug. 2011.

[66] V. Shivaldova, A. Paier, D. Smely, and C. F. Mecklenbräuker, "On Roadside Unit Antenna Measurements for Vehicle-to-Infrastructure Communications," in *Proc. IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2012.

[67] P. Greisen, S. Haene, and A. Burg, "Simulation and Emulation of MIMO Wireless Baseband Transceivers," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, 2010.

[68] D. Bates, S. Henriksen, B. Ninness, and S. R. Weller, "A 4x4 FPGA-based wireless testbed for LTE applications," in *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2008.

[69] K. Sobaihi, A. Hammoudeh, and D. Scammell, "FPGA Implementation of OFDM Transceiver for a 60GHz Wireless Mobile Radio System," in *Proc. International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, Dec. 2010.

[70] T. M. Fernández-Caramés, M. González-López, and L. Castedo, "Mobile WiMAX for Vehicular Applications: Performance Evaluation and Comparison Against IEEE 802.11p/a," *ELSEVIER Computer Networks Journal*, vol. 55, no. 16, pp. 3784–3795, Nov. 2011.

[71] H. Khaleel, F. Penna, C. Pastrone, R. Tomasi, and M. A. Spirito, "Frequency Agile Wireless Sensor Networks: Design and Implementation," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1599–1608, May 2012.

[72] F. Kaltenberger, R. Ghaffar, R. Knopp, H. Anouar, and C. Bonnet, "Design and Implementation of a Single-Frequency Mesh Network Using OpenAirInterface," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, 2010.

[73] A. Cipriano, P. Gagneur, A. Hayar, B. Zayen, and L. L. Floc'h, "Implementation and performance of an opportunistic cognitive radio system," in *Proc. Future Network and Mobile Summit*, Jun. 2010.

[74] Q. Wang, D. Fan, J. Chen, Y. Lin, Z. Zhu, and X. Dang, "WiMAX BS Transceiver Based on Cell Broadband Engine," in *Proc. IEEE International Conference on Circuits and Systems for Communications (ICCSC)*, May 2008.

[75] D. Kühling, A. Ibing, and V. Jungnickel, "12x12 MIMO-OFDM Realtime Implementation for 3GPP LTE+ on a Cell Processor," in *Proc. European Wireless Conference (EW)*, Jun. 2008.

[76] A. Ibing, D. Kühling, D. Wieruch, and H. Boche, "Software Defined Hybrid MMSE/QRD-M Turbo Receiver for LTE Advanced Uplink on a Cell Processor," in *Proc. IEEE International Conference on Communications Workshops (ICC Workshops)*, Jun. 2009.

[77] M. S. Khairy, C. Mehlführer, and M. Rupp, "Boosting Sphere Decoding Speed Through Graphic Processing Units," in *Proc. European Wireless Conference (EW)*, Apr. 2010.

[78] M. Wu, Y. Sun, and J. R. Cavallaro, "Reconfigurable Real-Time MIMO Detector on GPU," in *Proc. Asilomar Conference on Signals, Systems and Computers (ACSSC)*, Nov. 2009.

[79] A. Sghaier, S. Areibi, and R. Dony, "IEEE802.16-2004 OFDM Functions Implementation on FPGAs with Design Exploration," in *Proc. International Conference on Field Programmable Logic and Applications (FPL)*, Sep. 2008.

[80] A. A. Tabassam, F. A. Ali, S. Kalasit, and M. U. Suleman, "Building Software-Defined Radios in MATLAB Simulink - A Step Towards Cognitive Radios," in *Proc. UKSim Internacional Conference on Modelling and Simulation*, Apr. 2011.

[81] Y. A. Al-Zahrani, S. Al-Marshed, A. Al-Dhofyan, and A. I. Sulyman, "Design and FPGA Implementation of Reduced-Complexity MIMO-MLD Systems," in *Proc. IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Dec. 2010.

[82] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "An Architecture for Software Defined Cognitive Radio," in *Proc. AMC/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, Oct. 2010.

[83] Q. Wang, L. Zhuo, V. K. Prasanna, and J. Leon, "A Multi-Mode Reconfigurable OFDM Communication System on FPGA," in *Proc. Military and Aerospace Programmable Logic Devices (MAPLD)*, Jun. 2008.

[84] P. Murphy and A. Sabharwal, "Design, Implementation and Characterization of a Cooperative Communications System," *IEEE Trans. on Vehicular Technology*, vol. 60, no. 6, pp. 2534–2544, Jul. 2011.

[85] Z. Stamenkovic, K. Tittelbach-Helmrich, M. Krstic, J. I. nez, V. Elvira, and I. Santamaría, "MAC and baseband processors for RF-MIMO WLAN," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, 2011.

[86] X. Huang, C. Liang, and J. Ma, "System Architecture and Implementation of MIMO Sphere Decoders on FPGA," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 2, pp. 188–197, Feb. 2008.

[87] P. Suárez-Casal, A. Carro-Lagoa, J. A. García-Naya, and L. Castedo, "A Multicore SDR Architecture for Reconfigurable WiMAX Downlink," in *Proc. Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD)*, Sep. 2010.

[88] T. Haustein, A. Forck, H. Gäbler, V. Jungnickel, and S. Schiffermüller, "Real-Time Signal Processing for Multiantenna Systems: Algorithms, Optimization, and Implementation on an Experimental Test-Bed," *EURASIP Journal on Applied Signal Processing*, vol. 2006, 2006.

[89] V. Venkatkumar, T. Wirth, T. Haustein, and E. Schulz, "Relaying in Long Term Evolution: Indoor Full Frequency Reuse," in *Proc. European Wireless Conference (EW)*, May 2009.

[90] L. Thiele, V. Jungnickel, and T. Haustein, "Interference Management for Future Cellular OFDMA Systems Using Coordinated Multi-Point Transmission," *IECIE Trans. on Communications*, vol. E93-B, no. 12, pp. 3228–3237, Dec. 2010.

[91] V. Jungnickel, M. Schellmann, L. Thiele, and T. Wirth, "Interference-Aware Scheduling in the Multiuser MIMO-OFDM Downlink," *IEEE Communications Magazine*, vol. 47, no. 6, pp. 56–66, Jun. 2009.

[92] C. Schmidt-Knorreck, R. Pacalet, A. Minwegen, U. Deidersen, T. Kempf, R. Knopp, and G. Ascheid, "Flexible Front-End Processing for Software Defined Radio Applications Using Application Specific Instruction-Set Processors," in *Proc. Conference on Design and Architectures for Signal and Image Processing (DASIP)*, Oct. 2012.

[93] C. Schmidt-Knorreck, D. Knorreck, and R. Knopp, "IEEE 802.11p Receiver Design for Software Defined Radio Platforms," in *Proc. Euromicro Conference on Digital System Design (DSD)*, Sep. 2012.

[94] X. Chu and J. McAllister, "Software-Defined Sphere Decoding for FPGA-Based MIMO Detection," *IEEE Trans. on Signal Processing*, vol. 60, no. 11, pp. 6017–6026, Nov. 2012.

[95] P. Murphy, A. Sabharwal, and B. Aazhang, "On Building a Cooperative Communication System: Testbed Implementation and First Results," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, 2009.

[96] G. Wang, B. Yin, K. Amiri, Y. Sun, M. Wu, and J. R. Cavallaro, "FPGA Prototyping of a High Data Rate LTE Uplink Baseband Receiver," in *Proc. Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Nov. 2009.

[97] J. S. Park and T. Ogunfunmi, "Efficient FPGA-Based Implementations of MIMO-OFDM Physical Layer," *Circuits, Systems, and Signal Processing*, vol. 31, no. 4, pp. 1487–1511, Aug. 2012.

[98] V. P. Gil Jiménez, M. J. Fernández-Getino García, A. García Armada, R. P. Torres, J. J. García Fernández, M. P. Sánchez-Fernández, M. Domingo, and O. Fernández, "A MIMO-OFDM Testbed, Channel Measurements and System Considerations for Outdoor-Indoor WiMAX," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, 2010.

[99] X. Wu and J. S. Thompson, "FPGA Design of Fixed-Complexity High-Throughput MIMO Detector Based on QRDM Algorithm," in *Proc. International ICST Conference on Communications and Networking in China (CHINACOM)*, Aug. 2010.

[100] K. Amiri, J. R. Cavallaro, C. Dick, and R. M. Rao, "A High Throughput Configurable SDR Detector for Multi-User MIMO Wireless Systems," *Journal of Signal Processing Systems*, vol. 62, no. 2, pp. 233–245, 2011.

[101] V. Smolyakov, D. Patel, M. Shabany, and P. G. Gluak, "A WiMAX/LTE Compliant FPGA Implementation of a High-Throughput Low-Complexity 4x4 64-QAM Soft MIMO Receiver," in *Proc. Asilomar Conference on Signals, Systems and Computers (ACSSC)*, Nov. 2010.

[102] X. Hou, E. Zhou, J. Chen, Z. Zhang, and H. Kayama, "Robust Channel Estimator for MIMO-OFDM Systems with FPGA Implementation," in *Proc. Asia-Pacific Conference on Communications (APCC)*, Oct. 2008.

[103] Z. Iqbal and S. Nooshabadi, "Effects of Channel Coding and Interleaving in MIMO-OFDM Systems," in *Proc. IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2011.

[104] C. Sacchi, O. Tonelli, A. F. Cattoni, and Y. L. Moullec, "Implementation Aspects of a Flexible Frequency Spectrum Usage Algorithm for Cognitive OFDM Systems," in *Proc. IEEE Aerospace Conference*, Mar. 2011.

[105] S. Haene, D. Perels, and A. Burg, "A Real-Time 4-Stream MIMO-OFDM Transceiver: System Design, FPGA Implementation, and Characterization," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 6, pp. 877–889, Aug. 2008.

[106] D. L. Iacono, M. Ronchi, L. D. Torre, and F. Osnato, "MIMO OFDM Physical Layer Real-Time Prototyping," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2008.

[107] T. Onizawa, A. Ohta, and Y. Asai, "Experiments on FPGA-Implemented Eigenbeam MIMO-OFDM With Transmit Antenna Selection," *IEEE Trans. on Vehicular Technology*, vol. 58, no. 3, pp. 1281–1291, Mar. 2009.

[108] M. J. Canet, J. Valls, V. Almenar, and J. Marín-Roig, "FPGA Implementation of an OFDM-based WLAN Receiver," *ELSEVIER Microprocessors and Microsystems Journal*, vol. 36, no. 3, pp. 232–244, Feb. 2012.

[109] A. Recio and P. Athanas, "Physical Layer for Spectrum-Aware Reconfigurable OFDM on an FPGA," in *Proc. Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD)*, Sep. 2010.

[110] L. Boher, R. Rabineau, and M. Hélard, "FPGA Implementation of an Iterative Receiver for MIMO-OFDM Systems," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 6, pp. 857–866, Aug. 2008.

[111] A. Jiménez-Pacheco, A. Fernández-Herrero, and J. Casajús-Quirós, "Design and Implementation of a Hardware Module for MIMO Decoding in a 4G Wireless Receiver," *VLSI Design*, vol. 2008, 2008.

[112] K. Kokkinen, V. Turunen, M. Kosunen, S. Chaudhari, V. Koivunen, and J. Ryynänen, "FPGA Implementation of Autocorrelation-based Feature Detector for Cognitive Radio," in *Proc. NORCHIP*, Nov. 2009.

[113] S. Yoshizawa, K. Nishi, and Y. Miyanaga, "Reconfigurable Two-Dimensional Pipeline FFT Processor in OFDM Cognitive Radio Systems," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2008.

[114] M. S. Khairy, M. M. Abdallah, and S. E. D. Habib, "Efficient FPGA Implementation of MIMO Decoder for Mobile WiMAX System," in *Proc. IEEE International Conference on Communications (ICC)*, Jun. 2009.

[115] K. ElWazeer, M. M. Khairy, H. A. H. Fahmy, , and S. E. D. Habib, "FPGA Implementation of an Improved Channel Estimation Algorithm for Mobile WiMAX," in *Proc. International Conference on Microelectronics (ICM)*, Dec. 2009.

[116] S. S. Tehrani, S. Mannor, and W. J. Gross, "Fully Parallel Stochastic LDPC Decoders," *IEEE Trans. on Signal Processing*, vol. 56, no. 11, pp. 5692–5703, Nov. 2008.

[117] T. H. Pham, S. A. Fahmy, and I. V. McLoughlin, "Low-Power Correlation for IEEE 802.16 OFDM Synchronization on FPGA," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems (accepted)*, 2012.

[118] H. Kim, D.-U. Lee, and J. D. Villasenor, "Design Tradeoffs and Hardware Architecture for Real-Time Iterative MIMO Detection using Sphere Decoding and LDPC Coding," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 6, pp. 1003–1014, Aug. 2008.

[119] A. Burg, S. Haene, M. Borgmann, D. Baum, T. Thaler, F. Carbognani, S. Zwicky, L. Barbero, C. Senning, P. Greisen, T. Peter, C. Foelmi, U. Schuster, P. Tejera, and A. Staudacher, "A 4-Stream 802.11n Baseband Transceiver in 0.13μm CMOS," in *Proc. Symposium on VLSI Circuits*, Jun. 2009.

[120] M. Wenk, L. Bruderer, A. Burg, and C. Studer, "Area- and Throughput-Optimized VLSI Architecture of Sphere Decoding," in *Proc. IEEE/IFIP VLSI System on Chip Conference (VLSI-SoC)*, Sep. 2010.

[121] P. Maechler, P. Greisen, N. Felber, and A. Burg, "Matching Pursuit: Evaluation and Implementation for LTE Channel Estimation," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Jun. 2010.

[122] L. Bruderer, C. Studer, M. Wenk, D. Seethaler, and A. Burg, "VLSI Implementation of a Low-Complexity LLL Lattice Reduction Algorithm for MIMO Detection," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Jun. 2010.

[123] S. Yoshizawa and Y. Miyanaga, "VLSI Implementation of a 4x4 MIMO-OFDM Transceiver with an 80-MHz Channel Bandwidth," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2009.

[124] M. Hatanaka, R. Hashimoto, T. Tatsuka, T. Onoye, H. Hatamoto, S. Ibi, S. Miyamoto, and S. Sampei, "VLSI Design of OFDM Baseband Transceiver with Dynamic Spectrum Access," in *Proc. International Symposium on Intelligent Signal Processing and Communication Systems (IS-PACS)*, Dec. 2010.

[125] M. Ŝimko, D. Wu, C. Mehlführer, J. Eilert, and D. Liu, "Implementation Aspects of Channel Estimation for 3GPP LTE Terminals," in *Proc. European Wireless Conference (EW)*, Apr. 2011.

[126] C.-J. Huang, C.-W. Yu, and H.-P. Ma, "A Power-Efficient Configurable Low-Complexity MIMO Detector," *IEEE Trans. on Circuits and Systems I: Regular papers*, vol. 56, no. 2, pp. 485–496, Feb. 2009.

[127] G. Knagge, M. Bickerstaff, and B. Ninness, "A VLSI 8x8 MIMO Near-ML Detector with Preprocessing," *Journal of Signal Processing Systems*, vol. 56, no. 2, pp. 229–247, Jun. 2009.

[128] S. Saponara, N. E. LÍnsalata, and L. Fanucci, "Low-complexity FFT/IFFT IP Hardware Macrocells for OFDM and MIMO-OFDM CMOS Transceivers," *ELSEVIER Microprocessors and Microsystems Journal*, vol. 33, no. 3, pp. 191–200, May 2009.

[129] J. Löfgren, S. Mehmood, N. Khan, and B. Masood, "Hardware Implementation of an SVD Based MIMO OFDM Channel Estimator," in *Proc. NORCHIP*, Nov. 2009.

[130] C.-H. Yang and D. Markoviĉ, "A Flexible DSP Architecture for MIMO Sphere Decoding," *IEEE Trans. on Circuits and Systems I: Regular papers*, vol. 56, no. 10, pp. 2301–2314, Oct. 2009.

[131] J. Ketonen, M. Juntti, and J. R. Cavallaro, "Performance-Complexity Comparison of Receivers for a LTE MIMO-OFDM System," *IEEE Trans. on Signal Processing*, vol. 58, no. 6, pp. 3360–3372, Jun. 2010.

[132] P. Radosavljevic, K. J. Kim, H. Shen, and J. R. Cavallaro, "Parallel Searching-Based Sphere Detector for MIMO Downlink OFDM Systems," *IEEE Trans. on Signal Processing*, vol. 60, no. 6, pp. 3240–3252, Jun. 2012.

[133] M. Li, R. Appeltans, A. Amin, R. Torrea, H. Cappelle, M. Hartmann, H. Yomo, K. Kobayashi, A. Dejonghe, and L. V. D. Perre, "Overview of A Software Defined Downlink Inner Receiver for Category-E LTE-Advanced UE," in *Proc. IEEE International Conference on Communications (ICC)*, Jun. 2011.

[134] A. Alimohammad and B. F. Cockburn, "An Efficient Parallel Architecture for Implementing LST Decoding in MIMO Systems," *IEEE Trans. on Signal Processing*, vol. 54, no. 10, pp. 3899–3907, Oct. 2006.

[135] K. Mohammed and B. Daneshrad, "A MIMO Decoder Accelerator for Next Generation Wireless Communications," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 11, pp. 1544–1555, Nov. 2010.

[136] T. H. Yu, O. Sekkat, S. Rodriguez-Parera, D. Markoviĉ, and D. Ĉabrić, "A Wideband Spectrum-Sensing Processor With Adaptive Detection Threshold and Sensing Time," *IEEE Trans. on Circuits and Systems I: Regular papers*, vol. 58, no. 11, pp. 2765–2775, Nov. 2011.

[137] L. Zheng and D. N. C. Tse, "Diversity and multiplexing:A fundamental tradeoff in multiple-antenna channels," *IEEE Trans. on Information Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.

[138] G. Ganesan and P. Stoica, "Space-time block codes: a maximum SNR approach," *IEEE Trans. on Information Theory*, vol. 47, no. 4, pp. 1650–1656, May 2001.

[139] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *IEEE Trans. on Information Theory*, vol. 44, no. 2, pp. 744–765, Mar. 1998.

[140] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.

[141] T. K. Y. Lo, "Maximum ratio transmission," *IEEE Trans. on Communications*, vol. 47, no. 10, pp. 1458–1461, Oct. 1999.

[142] A. Pascual-Iserte, D. P. Palomar, A. I. A. I. Pérez-Neira, and M. A. Lagunas, "A robust maximin approach for MIMO Communications with imperfect channel state information based on convex optimization," *IEEE Trans. on Signal Processing*, vol. 54, no. 1, pp. 346–360, Jan. 2006.

[143] M. Payaró, A. Pascual-Iserte, and M. A. Lagunas, "Robust power allocation designs for multiuser and multiantenna downlink communication systems through convex optimization," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 7, pp. 1390–1401, Sep. 2007.

[144] G. L. Stuber, J. R. Barry, S. W. McLaughlin, Y. G. Li, M. A. Ingram, and T. G. Pratt, "Broadband MIMO-OFDM wireless communications," *Proc. IEEE*, vol. 92, no. 2, pp. 271–294, Feb. 2004.

[145] K. Schober, R. Wichman, and T. Koivisto, "MIMO adaptive codebook for closely soaced antennas arrays," in *Proc. European Signal Processing Conference (EUSIPCO)*, Sep. 2011.

[146] D. J. Love, R. W. Heath, and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2735–2747, Oct. 2003.

[147] T. Pande, D. J. Love, and J. V. Krogmeier, "Reduced Feedback MIMO-OFDM Precoding and Antenna Selection," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2284–2293, May 2007.

[148] L. Li, S. A. Vorobyov, and A. B. Gershman, "Transmit antenna selection based strategies in MISO communication systems with low-rate channel state feedback," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1660–1666, Apr. 2009.

[149] F. Mhiri, K. Sethom, and R. Bouallegue, "A survey on interference management techniques in Femtocell self-organizing networks," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 58–65, Jan. 2013.

[150] Ǐ. Güvenç, M.-R. Jeong, M. E. Şhain, H. Xu, and F. Watanabe, "Interference Avoidance in 3GPP Femtocell Networks Using Resource Partitioning and Sensing," in *Proc. International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC)*, Sep. 2010.

[151] J. Lotze, S. A. Fahmy, B. Özgül, J. Noguera, and L. E. Doyle, "Spectrum Sensing on LTE Femtocells for GSM Spectrum Re-Farming using Xilinx FPGAs," in *Proc. Software Defined Radio (SDR) Technical Conference and Product Exposition*, Dec. 2009.

[152] M. A. Abdelmonem, M. Nafie, M. H. Ismail, and M. S. El-Soudani, "Optimized Spectrum Sensing Algorithms for Cognitive LTE Femtocells," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, 2012.

[153] C. Bouras, G. Kavourgias, V. Kokkinos, and A. Papaziois, "Interference Management in LTE Femtocell Systems Using an Adaptive Frequency Reuse Scheme," in *Proc. Wireless Telecommunications Symposium (WTS)*, Apr. 2012.

[154] N. Saquib, E. Hossain, and D. I. Kim, “Fractional Frequency Reuse for Interference Management in LTE-Advanced HetNets,” *IEEE Wireless Communications Magazine*, Apr. 2013.

Reference literature

[WIM, 2005] (2005). IEEE 802.16e-2005. IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed Broadband Wireless Access Systems. Amendment 2: Physical and Medium Access Control Layer for Combined Fixed and Mobile Operation in Licensed Bands.

[LTE, 2010] (2010). 3GPP LTE Rel. 9. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation; 3GPP TS 36.211.

[(3GPP), 2009] (3GPP), T. G. P. P. (2009). Simulation assumptions and parameters for FDD HeNB RF requirements, 3GPP TSG RAN WG4 R4-092042.

[Amiri et al., 2012] Amiri, K., Duarte, M., Cavallaro, J. R., Dick, C., Rao, R., and Sabharwal, A. (2012). FPGA in Wireless Communications Applicatons (Ch. 3). In Rahman, A. and Anderson, J. H., editors, *FPGA Based Design and Applications, Integrated Circuits and Systems Series, Springer-Verlag GmbH, ISBN 978-0387726687.*

[Burg, 2006] Burg, A. (2006). *VLSI Circuits for MIMO Communication Systems.* PhD thesis, Eidgenössische Technische Hochschule (ETH) Zürich, Zürich (Switzerland).

[Burg et al., 2006] Burg, A., Belanovic, P., Felber, N., Lüthi, P., Studer, C., and Wenk, M. (2006). Algorithm Assessment Criteria for Hardware Implementation. In Burg, A., editor, *Deliverable D2.1.1, IST 26905 FP6 Multiple-Access Space-Time Coding Testbed (MASCOT) Project.*

[Caban, 2009] Caban, S. (2009). *Testbed-based Evaluation of Mobile Communication Systems.* PhD thesis, Technischen Universität Wien, Vienna (Austria).

[Caban et al., 2011] Caban, S., Mehlführer, C., Rupp, M., and Wrulich, M. (2011). *Evaluation of HSDPA and LTE: From Testbed Measurements to System Level Performance.* Wiley, ISBN 978-0-470-71192-7.

[Camera, 2008] Camera, K. B. (2008). *Efficient Programming of Reconfigurable Hardware through Direct Verification.* PhD thesis, University of California, Berkeley, California (United States of America).

[Deschamps et al., 2006] Deschamps, J.-P., Bioul, G. J. A., and Sutter, G. D. (2006). *Synthesis of Arithmetic Circuits - FPGA, ASIC, and Embedded Systems.* Wiley-Interscience, ISBN 978-0471-68783-2.

[Duarte, 2012] Duarte, M. (2012). *Full-duplex Wireless: Design, Implementation and Characterization.* PhD thesis, Rice University, Houston, Texas (United States of America).

[Eberle, 2006] Eberle, W. (2006). *Mixed Analog/Digital Exploration and Design for Wireless Boradband Transceivers.* PhD thesis, Katholieke Universiteit Leuven, Leuven, Vlaams-Brabant (Belgium).

[Eberli, 2009] Eberli, S. (2009). *Application-Specific Processor for MIMO-OFDM Software-Defined Radio*. PhD thesis, Eidgenössische Technische Hochschule (ETH) Zürich, Zürich (Switzerland).

[Edman, 2006] Edman, F. (2006). *Digital Hardware Aspects of Multiantenna Algorithms*. PhD thesis, Lund Uninversity, Lund, Skåne (Sweden).

[García-Naya et al., 2010] García-Naya, J. A., González-López, M., and Castedo, L. (2010). A Distributed Multilayer Software Architecture for MIMO Testbeds (Ch. 5). In Bazzi, A., editor, *Radio Communications, InTech, ISBN 978-953-307-091-9*.

[Gropp et al., 1999] Gropp, B., Lusk, R., and Skjellum, A. (1999). *Using MPI*. MIT Press, ISBN 0-262-57133-1.

[Haustein, 2006] Haustein, T. (2006). *Real Time Signal Processing for Multi-Antenna Systems and Experimental Verification on a Reconfigurable Hardware Test-bed*. PhD thesis, Technischen Universität Berlin, Berlin (Germany).

[ITRS, 2011] ITRS (2011). The International Technology Roadmap For Semiconductors, System Drivers, 2011 Edition.

[(ITU), 1996] (ITU), I. T. U. (1996). General requirements for instrumentation for performance measurements on digital transmission equipment, Rec. ITU-T O.150.

[(ITU), 1997] (ITU), I. T. U. (1997). Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000, Rec. ITU-R M.1225.

[ITU-R, 2003] ITU-R (2003). Rec. ITU-R M.1645: Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000.

[Kuo and Lee, 2001] Kuo, S. M. and Lee, B. H. (2001). *Real-Time Digital Signal Processing*. Wiley & Sons, ISBN 0-470-84137-0.

[Moore, 1965] Moore, G. E. (1965). Cramming More Components onto Integrated Circuits. *Electronics*, 38(8):114–117.

[Murphy, 2010] Murphy, P. O. (2010). *Design, Implementation and Characterization of a Cooperative Communications System*. PhD thesis, Rice University, Houston, Texas (United States of America).

[Naya, 2010] Naya, J. A. G. (2010). *Testbed Design for Wireless Communications Systems Assessment*. PhD thesis, Universidade da Coruña, A Coruña (Spain).

[Nilsson, 2007] Nilsson, A. (2007). *Design of Programmable Multi-Standard Baseband Processors*. PhD thesis, Linköping University, Linköping, Östergötland (Sweden).

[NTIA, 2011] NTIA (2011). United States Frequency Allocation Chart, National Telecommunications & Information Administration, United States Department of Commerce.

[Perels, 2007] Perels, C. D. (2007). *Frame-Based MIMO-OFDM Systems: Impairment Estimation and Compensation*. PhD thesis, Eidgenössische Technische Hochschule (ETH) Zürich, Zürich (Switzerland).

[Shannon, 1949] Shannon, C. E. (1949). Communication in the Presence of Noise. *Proc. Of The Institute of Radio Engineers (IRE)*, 37(1):10–21.

[Shariat et al., 2012] Shariat, M., Quddus, A. U., Bennis, M., Bharucha, Z., Lalam, M., Maqbool, M., Mayrargue, S., Kosta, C., De Domenico, A., Calvanese-Strinati, E., Mahapatra, R., de Lima, C. H. M., and Uygungelen, S. (2012). Promising Interference and Radio Management Techniques for Indoor Standalone Femtocells. In Quddus, A. U., editor, *Deliverable D3.2, ICT 248523 FP7 Broadband Evolved FEMTO Network (BeFEMTO) Project*.

[Studer et al., 2010] Studer, C., Wenk, M., and Burg, A. (2010). VLSI Implementation of Hard- and Soft-Output Sphere Decoding for Wide-Band MIMO Systems (Ch. 6). In Ayala, J. L., Alonso, D. A., and Reis, R., editors, *VLSI-SoC: Forward-Looking Trends in IC and Systems Design, Springer Berlin Heidelberg, ISBN 978-3-642-28565-3*.

[Wenk, 2010] Wenk, M. (2010). *MIMO-OFDM Testbed: Challenges, Implementations, and Measurement Results*. PhD thesis, Eidgenössische Technische Hochschule (ETH) Zürich, Zürich (Switzerland).

[XILINX®, 2012a] XILINX® (2012a). 7 Series DSP48E1 Slice, User Guide.

[XILINX®, 2012b] XILINX® (2012b). 7 Series FPGAs Configurable Logic Block (CLB), User Guide.

[XILINX®, 2012c] XILINX® (2012c). System Generator for DSP, User Guide.

[Zhang, 2001] Zhang, N. (2001). *Algorithm/Architecture Co-Design for Wireless Communications Systems*. PhD thesis, University of California, Berkeley, California (United States of America).

Internet sources

[ANA] *Analog Devices, VisualDSP++ Development Software.*
http://www.analog.com/VisualDSP.

[AQU] *Aquila, Open source DSP library for C++.*
http://aquila-dsp.org.

[ARR] *ArrayFire, High-level GPU matrix library.*
http://www.accelereyes.com/arrayfire/c.

[CAC] *Calypto®, Catapult® High-level Synthesis Tool.*
http://calypto.com/en/products/catapult/overview.

[CEL] *Cell SPE Task Library.*
http://cellspe-tasklib.sourceforge.net.

[DAC] *Texas Instruments, 16-bit 500 MSPS 2x-8x Interpolating Dual-Channel Digital-to-Analog Converter (DAC).* http://focus.ti.com/docs/prod/folders/print/dac5687.html.

[DCU] *Synopsys®, Design Compiler (DC) Ultra™.* http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DCUltra/Pages/default.aspx.

[DSP] *DSPFilters, A Collection of Useful C++ Classes for Digital Signal Processing.* https://github.com/vinniefalco/DSPFilters.

[DST] *MathWorks®, DSP System Toolbox™.* http://www.mathworks.com/products/dsp-system.

[EDI] *Cadence®, Encounter Digital Implementation System.* http://www.cadence.com/products/di/edi_system/pages/default.aspx.

[EMC] *MathWorks®, Embedded Coder®.* http://www.mathworks.com/products/embedded-coder.

[FRE] *Freescale™ Semiconductor, Complimentary ColdFire Digital Signal Processing Library (CFDSPLIB).* http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=CFDSPLIB.

[GED] *GEDOMIS® testbed.* http://engineering.cttc.es/gedomis.

[GEI] *GE Intelligent Platforms, AXISLib-GPU.* http://defense.ge-ip.com/products/axislib-gpu/p3547.

[HDC] *MathWorks®, HDL Coder™.* http://www.mathworks.com/products/hdl-coder.

[IPP] *Intel®, Integrated Performance Primitives (IPP).* http://software.intel.com/en-us/intel-ipp.

[MAT] *MathWorks®, MATLAB®.* http://www.mathworks.com/products/matlab.

[NVI] *NVIDIA®, GPU-Accelerated Libraries.* https://developer.nvidia.com/gpu-accelerated-libraries.

[NXP] *NXP Semiconductor, DSP library for LPC1700 and LPC1300 (Application Note).* www.nxp.com/documents/application_note/AN10913.pdf.

[PCT] *MathWorks®, Parallel Computing Toolbox™.* http://www.mathworks.com/products/parallel-computing.

[SCC] *Synopsys®, Synphony C Compiler.* http://www.synopsys.com/Systems/BlockDesign/HLS/Pages/SynphonyC-Compiler.aspx.

[SCD] *MathWorks®, Simulink Coder™.*
http://www.mathworks.com/products/simulink-coder.

[SIM] *MathWorks®, Simulink®.*
http://www.mathworks.com/products/simulink.

[SMC] *Synopsys®, Synphony Model Compiler.*
http://www.synopsys.com/Systems/BlockDesign/HLS/Pages/Synphony-Model-Compiler.aspx.

[SPT] *MathWorks®, Signal Processing Toolbox™.*
http://www.mathworks.com/products/signal.

[SPU] *SPUC, Signal processing using C++ - A DSP library.*
http://spuc.sourceforge.net.

[SYS] *XILINX®, System Generator for DSP™.*
http://www.xilinx.com/tools/sysgen.htm.

[TIC] *Texas Instruments, Code Composer Studio (CCStudio).*
http://www.ti.com/tool/ccstudio.

[TID] *Texas Instruments, SPRC121 TMS320C67x DSP Library.*
http://www.ti.com/tool/sprc121.

[VSG] *MathWorks®, Instrument Control Toolbox™.*
http://www.mathworks.com/products/instrument/hardware.

[VSL] *XILINX®, Vivado Electronic System Level (ESL) Design.*
http://www.xilinx.com/products/design-tools/vivado/integration/esl-design.

[YDL] *Yellow Dog Linux.*
http://www.yellowdoglinux.com.

Authored publications

[Bartzoudis 11] N. Bartzoudis, O. Font-Bach, A. Pascual-Iserte & D. López Bueno. *A Real-Time FPGA-based mobile WiMAX Transceiver Supporting Multi-Antenna Configurations.* In Proc. Argentine Conference on Micro-Nanoelectronics, Technology, and Applications (CAMTA), August 2011.

[Font-Bach 10] O. Font-Bach, N. Bartzoudis, A. Pascual-Iserte & D. López Bueno. *Design, Implementation and Testing of a Real-Time Mobile WiMAX Testbed Featuring MIMO Technology.* Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities (TridentCom), Revised Selected Papers, vol. 46, pages 199–208, May 2010.

[Font-Bach 11a] O. Font-Bach, N. Bartzoudis, A. Pascual-Iserte & D. López Bueno. *A Real-Time FPGA-Based Implementation of a High-Performance MIMO-OFDM Mobile WiMAX Transmitter.* Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT), Revised Selected Papers, vol. 81, pages 48–66, May 2011.

[Font-Bach 11b] O. Font-Bach, N. Bartzoudis, A. Pascual-Iserte & D. López Bueno. *A Real-Time MIMO-OFDM mobile WiMAX Receiver: Architecture, Design and FPGA Implementation.* ELSEVIER Computer Networks Journal, vol. 55, no. 16, pages 3634–3647, November 2011.

[Font-Bach 11c] O. Font-Bach, N. Bartzoudis, A. Pascual-Iserte & D. López Bueno. *Processing-Demanding Physical Layer Systems Featuring Single Or Multi-Antenna Schemes.* In Proc. European Signal Processing Conference (EUSIPCO), September 2011.

[Font-Bach 12a] O. Font-Bach, N. Bartzoudis, A. Pascual-Iserte & D. López Bueno. *A Real-Time FPGA-based Implementation of a High-Performance MIMO-OFDM Transceiver featuring a Closed-Loop Communication Scheme.* In Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), October 2012.

[Font-Bach 12b] O. Font-Bach, A. Pascual-Iserte, N. Bartzoudis & D. López Bueno. *MATLAB as a Design and Verification Tool for the Hardware Prototyping of Wireless Communication Systems (Ch. 9).* In V. N. Katsikis, editeur, MATLAB, A Fundamental Tool for Scientific Computing and Engineering Applications, Volume 2, InTech, ISBN 978-953-51-0751-4. 2012.

[Font-Bach 13a] O. Font-Bach, N. Bartzoudis, A. Pascual-Iserte, M. Payaró, L. Blanco, J. Serra & M. Molina. *An experimental real-time implementation of an interference management scheme in a LTE-based Macrocell-Femtocell HetNet scenario.* EURASIP Journal on Wireless Communications and Networking (to be submitted), June 2013.

[Font-Bach 13b] O. Font-Bach, N. Bartzoudis, M. Payaró & A. Pascual-Iserte. *Hardware-efficient implementation of a Femtocell/Macrocell interference-mitigation technique for high-performance LTE-based systems.* In Proc. International Conference on Field Programmable Logic and Applications (FPL), September 2013.