ANALYSIS OF SELECTED IMPLEMENTATION ISSUES OF A DYNAMIC SPECTRUM ALLOCATION ALGORITHM FOR OFDM SYSTEMS

APPLIED SIGNAL PROCESSING AND IMPLEMENTATION (ASPI)

Oscar Tonelli – Group 1047

AAU, 10th Semester, 2009

AALBORG UNIVERSITY

Institute of Electronic Systems
Fredrik Bajers Vej 7, A1
DK-9220 Aalborg East
Phone.: 9635 8600
Web: http://ies.auc.dk

TITLE:

Analysis Of Selected Implementation Issues Of a Dynamic Spectrum Allocation Algorithm For OFDM Systems.

PROJECT PERIOD:

1st February – 2nd June

SEMESTER:

Spring semester 2009

GROUP:

ASPI 1047

GROUP MEMBERS:

Oscar Tonelli

SUPERVISORS:

Yannick Le Moullec

Andrea Cattoni

COPIES:

4

Number Of Pages :

93

Supplement:

CD-ROM

Abstract

The next generation mobile communication systems, designed under IMT-A requirements, are expected to provide very high peak data rates. In order to reach such goal, 4G systems will require a very wide spectrum allocation. This kind of feature will lead to a different operational environment from today's systems. A possible scenario involves several operators sharing a common spectrum. Another operative scenario is given by a random and dynamic deployment context. It is therefore needed a method for resource allocation that ensures fair and effective spectrum use.
This work is intended to analyze the practical issues related to the implementation of a dynamic spectrum allocation algorithm and its physical feasibility on FPGA. The considered algorithm will be based on a WiMAX IEEE 802.16e-2004 OFDM PHY level in a simplified channel and transmission environment. After a first validation of the algorithm in a Matlab®-Simulink® environment, an analysis of the implementation cost in terms of area and execution time of selected algorithm functionalities on a Virtex 5 FPGA fitted on a Xilinx ML506 development board, is performed. Results indicates that the selected functionalities are not high demanding but the input ports requirements are a critic parameter,

PREFACE

This report serves as documentation for the master's thesis of group 09gr1047 in "Applied Signal Processing and Implementation" (ASPI) at the Institute of Electronic Systems of Aalborg University. The project is an "Analysis of Selected Implementation Issues Of A Dynamic Spectrum Allocation Algorithm For OFDM Systems".

Future mobile communication standards are expected to provide high data rates by using wide band channel allocations. Such wide allocations lead to new deployment scenarios where several operators are expected to share a common spectrum. The resource management among the active operators in the same area, becomes a key issue for the system development. Furthermore, the expectations about mobility features and network random deployment, push the research toward the development of dynamic spectrum allocation systems.
In this project an analysis of the implementation issues of a dynamic spectrum allocation algorithm is performed: at first the algorithm functionalities are individuated and implemented in Matlab/Simulink environment, in the second part of the project the system design in Xilinx System Generator is investigated in order to perform an analysis of the required hardware resources on an FPGA target platform.

This report is structured in five Chapters:

- Chapter 1 introduces the problem context, the adopted implementation methodology and the theory recalls used in the system development.
- Chapter 2 describes the adopted solution for the implementation of the simulation model in Matlab/Simulink and the implemented algorithm functionalities.
- Chapter 3 addresses the issues related to the implementation of specific functionalities with Xilinx System Generator.

- Chapter 4 presents the results of the performed simulations in Matlab/Simulink to validate the designed system, and the signals analysis of the System Generator implementation. The results of the resource estimation process performed with the System Generator compilation and ISE synthesis are then presented.
- Chapter 5 presents a summary of the project work and the conclusions. Additional future work and system optimization is also individuated.

In this project Mathworks Matlab/Simulink software is used for the implementation of a simulation model, and Xilinx System Generator is used to perform a hardware level implementation.

……

Oscar Tonelli

CONTENTS

SUMMARY

FIGURES

+

LIST OF ABBREVIATIONS

AMC Adaptive Modulation and Coding
AWGN Additive White Gaussian Noise
BER Bit Error Rate
BPSK Binary Phase Shift Keying
BS Base Station
DSP Digital Signal Processing
Demux Demultiplexer
FEC Forward Error Correction
FDD Frequency Division Duplexing
FFT Fast Fourier Transform
FPGA Field Programmable Gate Array
FSU Flexible Spectrum Usage
GSM Global System for Mobile Communications
HDL Hardware Description Language
HeNBs Home e-Node Base Stations
IEEE Institute of Electrical and Electronics Engineers
IFFT Inverse Fast Fourier Transform
IMT-A International Mobile Telecommunications-Advanced
ISE Integrated Software Environment
ISI Inter Symbol Interference
LA Local Area
LI Low Interference
LOS Line-Of-Sight
MATLAB Matrix Laboratory
Mbps Megabit per second
MIMO Multiple Input – Multiple Output
MISO Multiple Input – Single Output
MUX Multiplexer

NLOS Non Line-Of-Sight
OFDM Orthogonal Frequency Division Multiplexing
OFDMA Orthogonal Frequency Division Multiple Access
PHY Physical Layer
PRB Physical Resource Block
PSK Phase Shift Keying
QAM Quadrature Amplitude Modulation
QPSK Quadrature Phase Shift Keying
RAM Random Access Memory
SINR Signal to Interference and Noise Ratio
SLB Spectrum Load Balancing
SNR Signal to Noise Ratio
SS Subscriber Station
TDD Time Division Duplexing
UMTS Universal Mobile Telephony System
VHDL VHSIC Hardware Description Language
WiMAX Worldwide Interoperability for Wireless Microwave Access
WiFi Wireless Fidelty
WF Water Filling

1. PROBLEM ANALYSIS

1.1 INTRODUCTION

Communication is a fundamental activity for all human beings; since the very first ages of mankind, communicating with other humans has always been a primary need to survive in an organized community. Communicating in all the possible forms (speech, writing, visual signs, etc.) it is not only a necessity to satisfy material issues but also a kind of activity that we could summarize as expression of human sensitivity.

The effectiveness and power of the message we would like to communicate often relies on the technology we are using; it is easy to understand how the invention of printing for example, gave a huge boost to the diffusion of ideas and thoughts, or how the radio telecommunications have been able to connect people all around the globe.

In the last century we have witnessed an impressive development in the communication technology due to intensive research in radio propagation, electronics and computer networks. Starting from early radio or television broadcasting, several new communication services have been developed. At present day, for instance, the use of mobile phones sending SMS and MMS, or domestic high speed internet connections are established technologies.

In modern society, telecommunication technologies do not simply provide tools to carry a message, but have gained a fundamental role in stimulating work productivity and allowed the creation of new business applications. The strictly connection between the

specific technology used and business, stimulated the research to overcome physical restrictions that limit the diffusion of new services.
Through the years, several telecommunication systems, focusing their attention on major issues like mobility, inter-connectivity and the enhancement of data-rate transmission have been developed. Standards like GSM, UMTS, WiFi or WiMAX, addressed these issues from different perspectives, developing specific features for different usage context.
We might expect future research to proceed towards the development of standards that enable all the desired features in a single technology.

1.1.1 COMMUNICATION STANDARDS DEVELOPMENT

Some very interesting considerations about the development of the wireless technology are reported in [3]. Here are individuated four main principles that guide the progress of the communication technology and the standardization process.

1) Shannon's law [6]:

$$W\ log_2\left(1+\frac{P}{N_0 W}\right) \qquad (1.1.1.1)$$

Where W is the signal bandwidth in Hz, P is the power of the signal in Watts, N_0 is the one-sided noise power spectral density in W/Hz.

In case of one transmitter and one receiver, the communication channel capacity depends linearly on the available signal bandwidth and logarithmically on the Signal to Noise Ratio (SNR): it is therefore easier to achieve a greater capacity increasing the bandwidth than the SNR.

2) Moore's law [7] that states the level of integrations of integrated circuits to be doubled every 18 months: this provides the foundation to the development of increasingly complex and powerful signal processing circuits.

3) The value of a network is proportional to the square of the connection speed (R.Metcalfe) [8].

4) The value of a network is proportional to the square of the number of connected devices [8].

These principles address the question of what can be accomplished in a communication system and how. Basically the efforts go in the direction of increasing the rate of the communication and increasing the number of connected devices. The development of telecommunication standards as the IEEE 802 family, follows these guidelines and the approved wireless standards such as the 802.11, the 802.16 and amendments, provide the technology to interconnect a large number of devices and increasing data rates.

The characteristics of an ideal wireless technology can be summarized [3] in :

- Ability to transport real-time voice, audio, video and high computer data-rate.
- Allow devices with different features to inter-operate together.
- Allocating the spectrum among the connected devices in a efficient and dynamic way.

The business and market requirements historically pushed the development of voice-oriented networks first, and data-oriented after, following two separate paths. At present the long-term market trend is to merge the technologies, obtaining common platforms for the development of new services.

1.2 Project Context

The development of a technology, especially in a business context is a combination of two factors: the requests from the users and the possibility of delivering new typologies of services given by the technology itself. We may consider for instance, a home user expectations: basic office applications and multimedia services will be the primary requests, current standards already address them: other features as permanent connectivity and multiplatform compatibility instead, would be the new factors that boost the quality of service and lead the user, in a completely new era of telecommunication systems giving the chance of offering new typology of commercial services.

Current research on communication technology and new wireless standards, is actually exploiting this kind of features, investigating new solutions for enhancing mobility, flexibility, and performances of the next generation systems.

One of the most interesting topics is related to the possibility of having even more self-managed systems, where the interconnected devices can autonomously share the

available resources and perform the necessary network reconfigurations in case of dynamic environments. In particular one of the key problems is the channel allocation among different transmitting systems: this project analyses and investigates issues related to the implementation of a dynamic spectrum allocation algorithm.

1.3 Problem Statement

This work is intended to analyze the issues related to the implementation of the main functionalities of a dynamic spectrum allocation algorithm, and to investigate its physical feasibility on an FPGA platform. The question this project attempts to answer is:

"*What are the implementation key challenges related to the feasibility analysis of a modified dynamic spectrum algorithm for an OFDM system?"*

1.4 Problem Scope

The project is focused on the resource allocation in a wireless system, based on a OFDM transmission scheme; the first goal is to evaluate how the algorithm works in a simulated environment and what are the main issues that concerns its utilization. The considered algorithm is based on a WiMAX IEEE 802.16-2004 OFDM PHY level [9] and the environment modeled with specific limitations and assumptions. After a first validation of the algorithm in a Matlab®-Simulink® environment [10], an analysis of the implementation cost in terms of area and execution time of selected algorithm functionalities on a Virtex 5 [11] FPGA fitted on a Xilinx ML506 [12] development board is performed.
The implementation of a full detailed environment is not within the scope of this project: the features considered are limited to provide the necessary support for the algorithm testing.
The deployment settings consider three Base Stations (BS) placed in different fixed positions sharing a common channel. The simulations consider a fixed transmission context where the three BS are simultaneously activated.
A more detailed description about implementation assumptions and limitations is presented in Chapter 2.

1.5 METHODOLOGY

In this project, the implementation platform is HW-oriented (and is based on FPGA). In order to design and implement efficiently the dynamic spectrum allocation algorithm, it is highly desirable to follow a well-defined methodology. This is discussed in what follows.

1.5.1 DESIGN PROCESS

System design is the process that consists in the development of a certain number of functions on a silicon chip [13]; the methodology mainly defines the models and the transformations that lead from the specifications to the proper implementation of the system. The conceptual framework provided by a well-structured methodology helps to conceptualize, categorize and visualize the design automation issues.
The complexity of the design process can be handled by defining three main concepts, namely, *hierarchy*, *abstraction* and *domains* [14].

- HIERARCHY
  Hierarchy is a partitioning of the design model, that hides the details of a part in a lower hierarchical level. The purpose of the hierarchy is to hide information when not needed and show the details when they are useful.

- ABSTRACTION
  Abstraction focuses on the typology of the information provided to represent a system: at different levels correspond different modeling concepts and used semantics. The movement from different abstraction levels implies a decision making process. In the design development towards a low level implementation more abstract models are replaced by less abstract models. For example at system-level only few blocks describe input/output relations in a fast but not detailed manner, at the transistor-level instead, millions of blocks describe the behavior of each transistor, providing a very detailed description but slow to perform.

- DOMAINS
  Individuating different domains allow to focus on one design aspect per time. A domain is an aspect of a model that can be analyzed independently from others.
  Example of domain are data, time, communication or computation.

In a top-down approach, the system design process starts at high level of abstraction with a behavioral description; through gradual steps it is progressively refined and more details are added to perform the implementation. In a bottom-up approach, “simple” blocks are combined to create more and more complex systems. An efficient implementation of the algorithm can be achieved exploring the implementation space on multiple abstraction levels: the design model enables this activity providing a systematic flow to the system design.
The model specifies the functionalities of the system that can be described in detail; more specifically this project utilizes the Rugby Meta-model as framework for the design process, in a top-down fashion.

1.5.2 The Rugby Meta- Model

The Rugby meta-model [14] is a conceptual framework based on the previous definitions: four domains are identified : *computation*, *communication*, *time* and *data*.
The aim of Rugby meta-model is to overcome the limitations of older models, like the Y-charts [15], by dealing with concurrent processes and mixed hardware/software implementations. The name derives from the similarity to a rugby, when the domain lines that proceed from the starting point (Idea) to the final implementation (Physical system), are visualized as depicted in Figure 1.

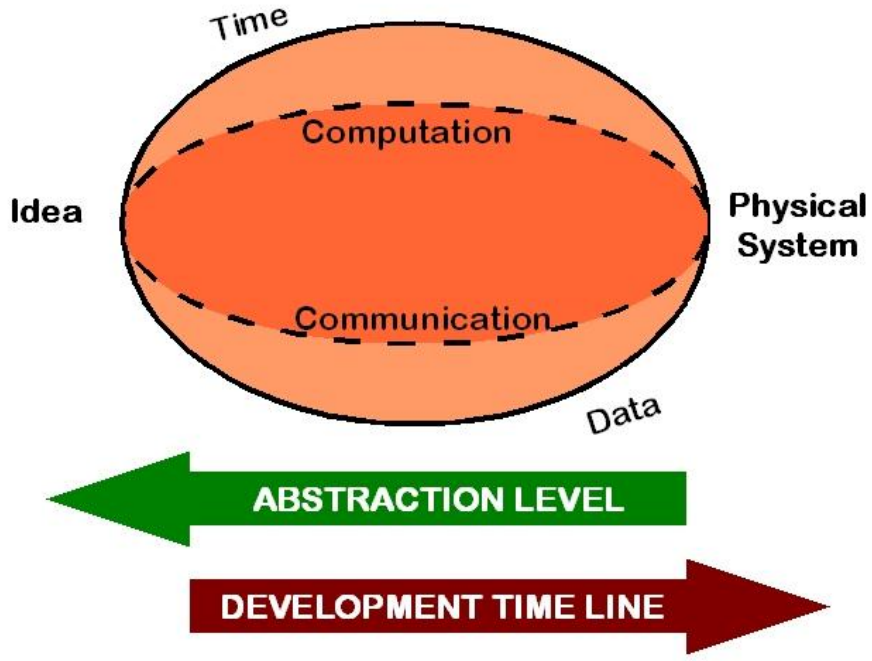

Figure 1 - The Rugby Meta-Model

STARTING FROM THE INITIAL IDEA THE MODEL VISUALIZES THE DEVELOPMENT PROCESS THAT LEADS TO THE PHYSICAL IMPLEMENTATION EXPLORING TIME, COMPUTATION, COMMUNICATION, DATA DOMAINS.

The concept of the Rugby meta-model is to allow a parallel analysis of the four domains during the development time. The design process starts with a higher level of abstraction, that is the original idea, and progressively reduces the abstraction to focus on more physical details in order to accomplish a physical implementation. In addition, the Rugby meta-model allows the analysis of both hardware or/and software oriented applications, and therefore the domain structures can split in two different design specializations.

Figure 2 provides an example of the Rugby meta-model domain line (here the communication domain): decreasing the abstraction level, the domain line splits depending on hardware and/or software implementation.

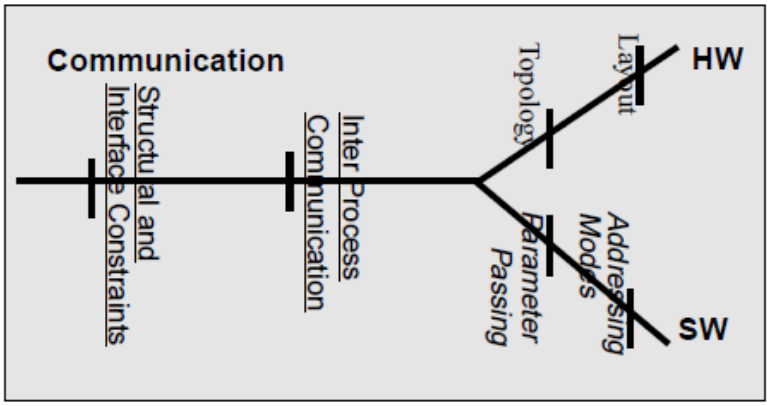

FIGURE 2 EXAMPLE OF ABSTRACTION LEVELS IN THE COMMUNICATION DOMAIN [14]

The characteristics of the Rugby meta-model domains [14] are now briefly described.

- COMPUTATION
  The computation domain is focused on the relationship between the input-output values at different abstraction levels. Different models are used at different levels, from Relations and Constraints at highest level, down to Transistors at lowest level. At the Logic Block level, models are based on boolean functions, at the transistor level instead, the differential equations representing the V-I characteristics are considered.

- COMMUNICATION
  The communication domain deals with the connections and interactions between the various design elements (Figure 2). At high level are considered interfaces

and communication constraints. At lower level like Layout, which is based on geometrical principles, physical units are used to describe geometrical parameters. For software implementation the focus is on data addressing and control buses.

- DATA
  The Data domain is concerned with data types and data objects that are managed by the design entities: the abstraction levels start from the Data Constraints down to the analog values or the Data type managed by the processor.

- TIME
  Time is another parameter that requires a deep analysis in system design: at different levels of abstraction the "time concept" is different. The Rugby meta-model Time domain considers the various steps that lead from the time constraints and the causality relations, to the physical time and propagation delay of the hardware devices.

1.5.3 PROJECT MAPPING ON THE MODEL

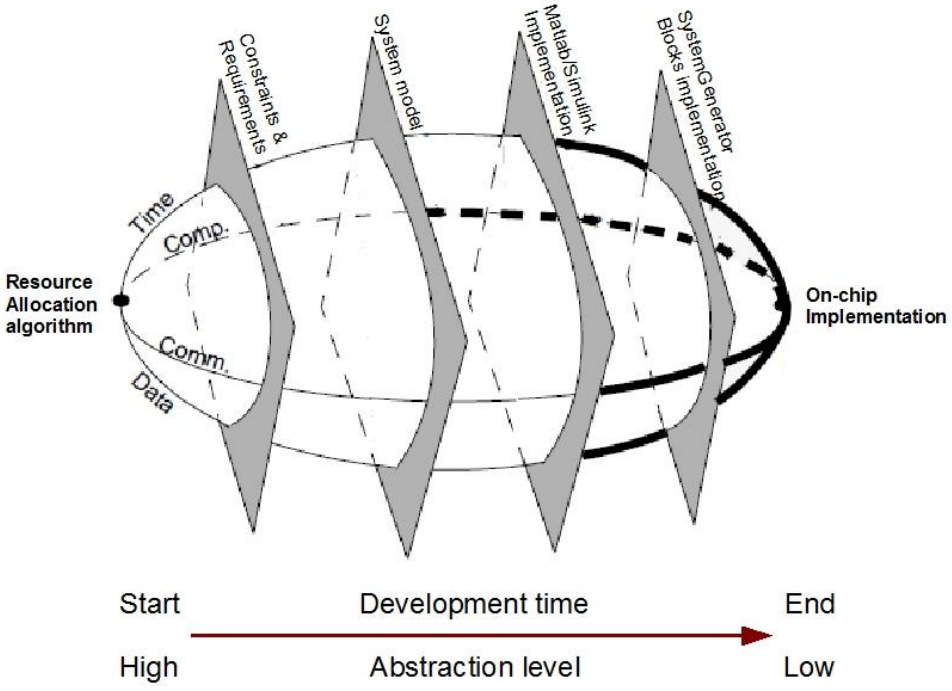

FIGURE 3 – PROJECT ABSTRACTION LEVELS MAPPED INTO THE RUGBY META-MODEL SCHEME PROPOSED IN [14]

The Rugby meta-model is used as model design reference in this project. In figure 3 the different abstraction levels of the project development are mapped in the original Rugby model. The implementation process moves from the initial Idea (implementation of a resource allocation algorithm) to the final implementation on FPGA board ("On-chip" in Figure 3).
The four rugby domains are exploited through different abstraction levels, in this project four main levels have been individuated:

- At first the constraints and requirements are individuated; the desired system should implement a dynamic resource allocation algorithm inspired by [2] and presented in section 1.9. Assumptions and delimitations of the project scope are also defined.
- In the system model, the necessary structure for the algorithm implementation is defined: the algorithm operates on an OFDM-based communication system in a specific transmission context where three stations are transmitting simultaneously on a shared channel.
- The third abstraction level is represented by the Matlab/Simulink implementation: at this level the complete simulation environment where the algorithm operates, is implemented. The algorithm functionalities are implemented with Matlab/Simulink blocks and functions in order to test the adapted algorithm validity.
- After having individuated the necessary blocks that define the algorithm functionalities, the key blocks are implemented with Xilinx System Generator blocks to a lower level of abstraction (Register Transfer Level (RTL), VHDL).

The System Generator structure allows finally to analyze the physical requirements of the blocks and to download the firmware on the FPGA board.

1.6 OFDM

1.6.1 INTRODUCTION

In this project an algorithm for dynamic resource allocation is considered . As reported later in Section 1.9, the algorithm needs to rely on a system able to provide the necessary flexibility in terms of resource allocation over time and frequency. Orthogonal Frequency

Division Multiplexing (OFDM) is considered a suitable candidate for its characteristic of multicarrier transmission.
The following section briefly recalls the theory principles that describe the functioning of an OFDM system .

1.6.2 Theory Principles

This section is based on the OFDM General Principles presented in [16].

The OFDM transmission is the implementation of the multicarrier transmission concept.
In a digital transmission scheme with a linear modulation (M-PSK, M-QAM) the occupied bandwidth is given by :

$$B = (1+\alpha)T_s^{-1} \qquad (1.6.2.1)$$

where T_s is the symbol duration and $\propto$ is the roll-off factor of a raised-cosine pulse. The reception on a channel with delay spread τ_m incurs in Intersymbol Interference (ISI) if $\tau_m \gg T_s$; the maximum allowed bit rate, is therefore limited by the delay spread of the channel :

$$R_b = \log_2(M)T_s^{-1} \qquad (1.6.2.2)$$

where M is referred to the modulation.
A simple method to overcome this kind of limitation is to split the data stream into K substreams with lower data rate and transmit all the streams on adjacent subcarriers. This actually consists in a parallelization in the frequency domain that does not affect the total bandwidth required for the transmission.

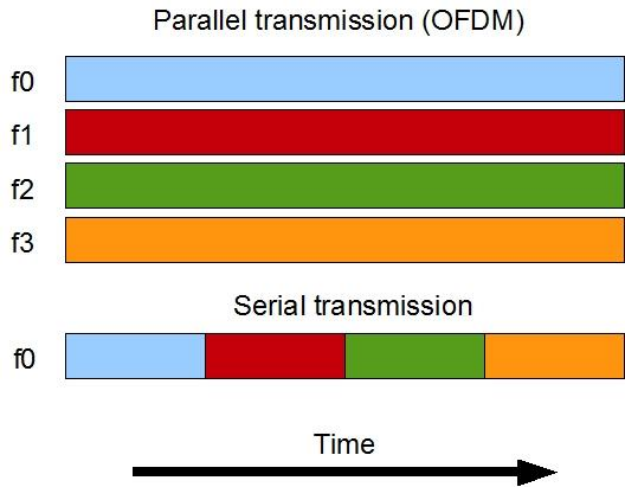

Figure 4 – Multicarrier transmission concept.
f0, f1, f2, f3 are the sub-carrier frequencies, in serial transmission only one carrier is used.

The bandwidth on the single subcarrier is B/K and the symbol duration T_s is increased by K, allowing a K-times higher data rate for the same delay spread factor. Although splitting the stream on the subcarriers can improve the total data rate, too long symbol duration make the transmission sensitive to the channel time incoherence. The maximum Doppler frequency ν_{max} defines then another condition that must be fulfilled for correct transmission [16] :

$$\nu_{max} T_s \ll 1 . \qquad (1.6.2.3)$$

It is possible to individuate two major schemes for the multicarrier idea implementation; the first one is modulating independently K individual carriers (Figure 5) while the second one is based on the utilization of a filter bank of K bandpass filters excited by a parallel data stream (Figure 6). The last mentioned concept is usually preferred in practical systems implementation [16].

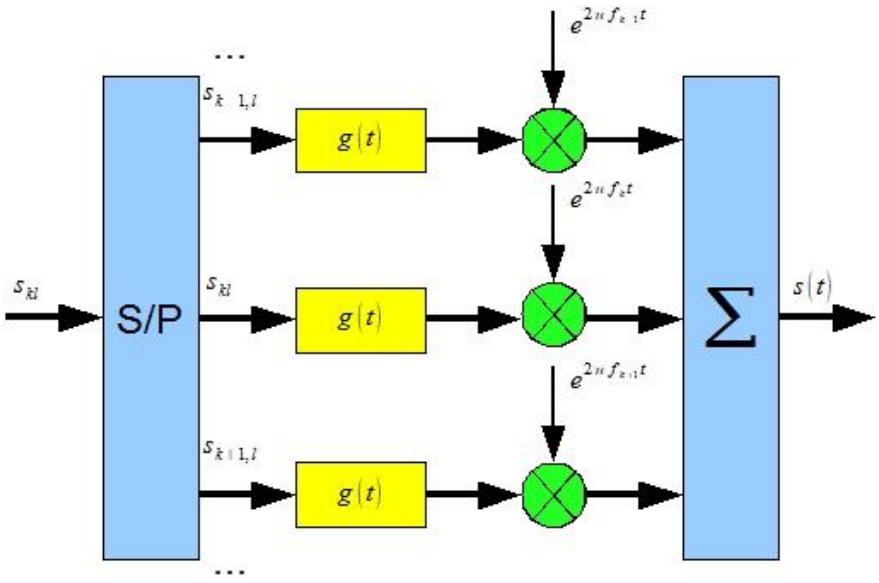

FIGURE 5 – BLOCK DIAGRAM FOR MULTICARRIER SCHEME IMPLEMENTATION (1) IN THIS SCHEME THE K CARRIERS ARE MODULATED INDEPENDENTLY [16].

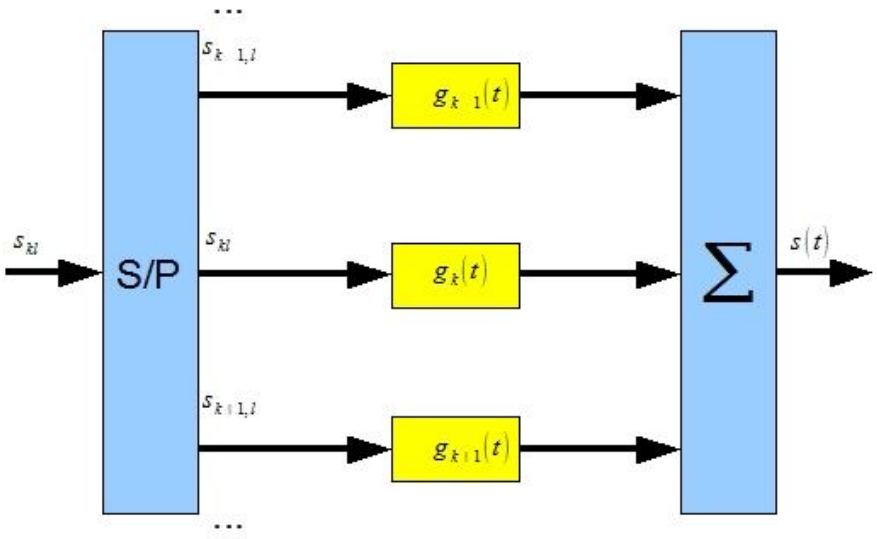

FIGURE 6 – BLOCK DIAGRAM FOR MULTICARRIER SCHEME IMPLEMENTATION (2)
IN THE SECOND SCHEME A FILTER OF K BANDPASS FILTERS IS EXCITED BY THE PARALLEL DATA STREAM [16]

Let us now consider the second implementation scheme depicted in Figure 6. We have K different transmit pulses $g_k(t)$ located at $f = f_k$: the parallel data stream excites a filter bank of different bandpass filters. The outputs are then summed before the transmission.
In a compact expression we obtain:

$$s(t) = \sum_{kl} s_{kl} g_{kl}(t) \qquad (1.6.2.4)$$

The OFDM orthogonality is strictly related to the chosen base transmission pulse $g(t)$; the chosen pulse bases should fulfill the following orthogonality property [16]:

$$\langle g_{kl}, g_{k'l'} \rangle = \delta_{kk'} \delta_{ll'} \qquad (1.6.2.5)$$

Two pulses are always orthogonal if they do not overlap either in time or frequency domain; that pulse cannot be limited in both domains at the same time so a precise choice has to be made.
What in practical implementation is commonly considered are band limited pulses orthogonal in time; in particular square root raised-cosine pulses. We previously defined the bandwidth of such typology of pulses, when the roll off factor α is equal to zero we obtain and ideal rectangular spectral shape and sinc shaping in time domain.
In Figure 7 a spectral representation of a typical OFDM modulation is depicted.

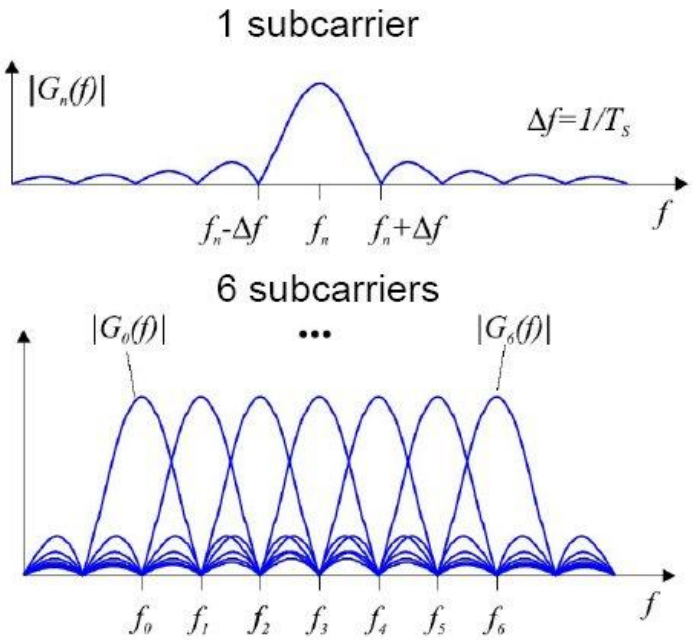

FIGURE 7 – OFDM SUBCARRIERS WITH TYPICAL OVERLAPPING PULSES SHAPE [4]

1.6.3 FFT IMPLEMENTATION

The implementation of an OFDM with Fourier base pulses and a rectangular shape of length T_s in time, is quite simple and it is commonly preferred for practical purposes [16].
In one time interval the transmit signal is :

$$s(t) = \frac{1}{\sqrt{T}} \sum_{k=-K/2}^{K/2} s_k \exp\left(j2\pi \frac{k}{T} t \right) \Pi\left(\frac{t}{T} - \frac{1}{2} \right) \qquad (1.6.3.1)$$

Where K is the subcarriers number, s_k the data symbols.
To recover the data symbols s_k, the perfectly synchronized receiver performs a Fourier analysis:

$$s_k = \langle g_k, s \rangle = \frac{1}{\sqrt{T}} \int_0^T \exp\left(- j2\pi \frac{k}{T} t \right) s(t) dt \,. \qquad (1.6.3.2)$$

The analysis is usually performed by means of a Fast Fourier transform (FFT) while the synthesis by the Inverse Fast Fourier Transform (IFFT): the stream of digitally modulated symbols is divided into blocks of length K (or K+1), discretely transformed by the IFFT, digital-analog converted and finally transmitted.
Figure 8 shows the complete transmitter/receiver OFDM implementation by FFT.

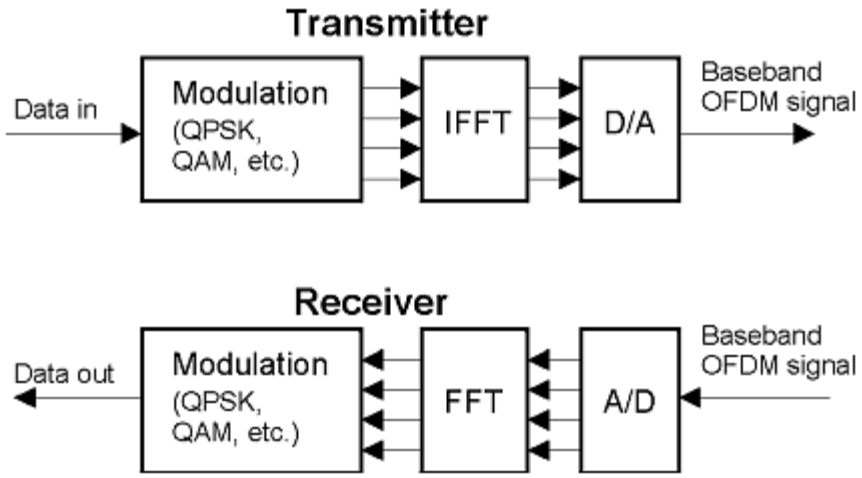

FIGURE 8 - IMPLEMENTATION OF OFDM WITH FFT

THE BLOCKS ON THE TRANSMITTER SIDE HAVE THE CORRESPONDING INVERSE AT RECEIVER SIDE.

The size of the FFT is a key element in the OFDM system design, since a large FFT size would increase the symbol time and offer better protection against multipath delay spread, but a reduced subcarrier spacing makes the system more vulnerable to the intercarrier interference given by the Doppler Spread in mobile application [5].

1.6.2 CONSIDERATIONS ON OFDM

The OFDM system is a good solution for many communication purposes because of its many advantages [5]:

- **Low computational complexity:** the computational complexity of an FFT/IFFT system is much lower than a standard equalizer-based system.
- **Slow degradation of performance in case of excess delay:** greater classification and low constellation sizes provide fallback rates more robust against the delay spread. OFDM is well suited for adaptive modulations that allows the system to make the best of the available channel condition.
- **Robustness against deep fades:** the use of separate subcarriers provides robustness against burst errors and fades of the channel.
- **Robust against narrowband interference:** such interference affects only a fraction of subcarriers.

- **Suitable for coherent demodulation:** a pilot based channel estimation is quite easy to implement in a OFDM system, enabling coherent demodulation (more power efficient).
- **Multi access scheme:** OFDM can easily be adapted for a multi access scheme where different users can be allocated on different subcarriers and individual adaptive modulations adopted in order to enhance the total capacity.

OFDM however shows some challenges, especially related to the high peak-to-average ratio that may cause non-linearities and distortions. In addition a good frequency synchronization is required to avoid phase noise and frequency dispersion.

1.7 WiMAX-802.16

1.7.1 Overview

In this project, a simulation model based on the WiMAX IEEE 802.16-2004 OFDM PHY level [9] is used. Despite the choice of the simulation model has been performed on the available simulation features rather than the WiMAX environment characteristics, a better understanding of the technology is needed in order to better exploit the model features. In the following section, the main characteristics of the WiMAX technology are described.

WiMAX is a wireless digital communications system, also known as IEEE 802.16, that is intended for wireless "metropolitan area networks". The purpose of WiMAX networks is to provide a wide broad band connection on a wireless infrastructure.
Since the deployment costs and efforts for wide area communication infrastructure, commonly wire-based, are often exorbitant, the WiMAX technology aims to provide a more easy-deployable and less expensive solution to achieve a broadband network coverage.
The characteristics of the 802.16 systems improve, in addition, the wireless experience provided by the 802.11 (WiFi) technology. WiMAX offers a point-to-point range of 50Km in Line-Of-sight (LOS), and supports a large number of connected devices with an increased tolerated multipath-signal delay that grants a better mobility.
During the standardization process, several amendments introduced new features to the original 802.16 standard version (2001). In 2004 the revised IEEE 802.16-2004 was presented. This standard targets fixed applications and is often referred as “fixed

WiMAX". Later in December 2005 the new IEEE 802.16e-2005 added the mobility support to the previous standard and is referred as "mobile WiMAX".
The standards offer a variety of design options and suits several applications and deployment scenarios. For practical reason related to interoperability of commercial implementations, a limited number of system profiles and certification profiles have been defined by the WiMAX Forum. Some features in system profiles can be different from what is in the original IEEE standard.
The WiMAX Forum has two different system profiles [5]: one based on IEEE 802.16-2004, OFDM PHY, called the fixed system profile; the other one based on IEEE 802.16e-2005 scalable OFDMA PHY, called the mobility system profile.
In this project the WiMAX simulation model is based on a model that supports the 802.16-2004 profiles defined by the WiMAX Forum. The starting simulation model has been then modified in order to support the algorithm testing.

In the following section of the report, WiMAX features relevant to the certification profiles and to the utilized model, are described.

1.7.2 Main features of WiMAX

This sub-section builds upon [4] and [5].

- OFDM
  The use of the Orthogonal Frequency Division Multiplexing (OFDM) is one of the main features of WiMAX. OFDM is widely used for its resistance to interference and signal degradation (multipath, delay spread) and allows WiMAX to operate in NLOS conditions.
- TDD and FDD
  WiMAX supports both Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD). TDD is preferred in most of the implementations because of its flexibility in managing different data rates in uplink-to-downlink connections and less complexity in transceiver design.
- Adaptive Modulation and coding (AMC)
  WiMAX supports several modulation and forward error correction (FEC) schemes, and allow the schemes to be changed on a per user or per frame basis based on channel conditions. The AMC system is an effective way to maximize the throughput on a time-varying channel. The scheme choice is made evaluating the Signal to Noise and Interference Ratio at the receiver, for each user.

- ORTHOGONAL FREQUENCY DIVISION MULTIPLE ACCESS (OFDMA)
  The different users can be allocated in different subsets of OFDM tones: this multiple access technique exploits the frequency diversity and multiuser diversity allowing improvements in the system capacity. In this project the OFDMA functionalities are not properly implemented since the management of multiple user transmission on a single base station is not considered. It is, however, possible to introduce the OFDMA functionalities without major modifications to the model, since the underlying principles remain valid.

1.8 OFDM IN WIMAX PHY

Different versions of WiMAX have different OFDM implementations [5], with different supported channel bandwidths or scalable OFDM with 128, 512, 1,024, 2,048 subcarriers in the IEEE 802.16e-2005. Fixed WiMAX, based on the IEEE 802.16-2004 considered in this project, uses a 256 points FFT-based physical OFDM layer and a 3.5 MHz channel bandwidth.
In the considered Fixed WiMAX profile, the size of the FFT is fixed at 256: 192 subcarriers are used to carry data, 8 pilots are reserved for channel estimation and synchronization, and the rest for guard band subcarriers. With a fixed size FFT, the subcarrier spacing varies with the channel bandwidth: increasing the bandwidth subsequently increases subcarrier spacing and reduces symbol time. In order to overcome the delay spread, new guard time configurations have to be considered. WiMAX indeed, allows several guard times to be used in a proper trade-off consideration between spectral efficiency and delay spread robustness: in good multipath conditions a 3% guard time can be considered sufficient, a 25% time ensure maximum delay spread robustness.

1.8.1 ADAPTIVE MODULATION AND CODING IN WIMAX

Depending on channel conditions, WiMAX supports several modulation and coding schemes that can change on a burst-by-burst basis. This feature is implemented by using a channel quality feedback indicator provided by the user in uplink to the BS. The base station scheduler can manage different channel quality parameters for each user, and set the proper modulation and coding scheme that maximizes the throughput for the available signal-to-noise ratio.

Table 1 shows the supported Modulation and Coding features.

	Downlink	Uplink
Modulation	BPSK, QPSK, 16 QAM, 64 QAM; BPSK optional for OFDMA-PHY	BPSK, QPSK, 16 QAM; 64 QAM optional
Coding	Mandatory: convolutional codes at rate 1/2, 2/3, 3/4, 5/6 Optional: convolutional turbo codes at rate 1/2, 2/3, 3/4, 5/6; repetition codes at rate 1/2, 1/3, 1/6, LDPC, RS-Codes for OFDM-PHY	Mandatory: convolutional codes at rate 1/2, 2/3, 3/4, 5/6 Optional: convolutional turbo codes at rate 1/2, 2/3, 3/4, 5/6; repetition codes at rate 1/2, 1/3, 1/6, LDPC

TABLE 1 - SUPPORTED MODULATION AND CODING IN WIMAX [5]

1.9 THE RESOURCE ALLOCATION ALGORITHM

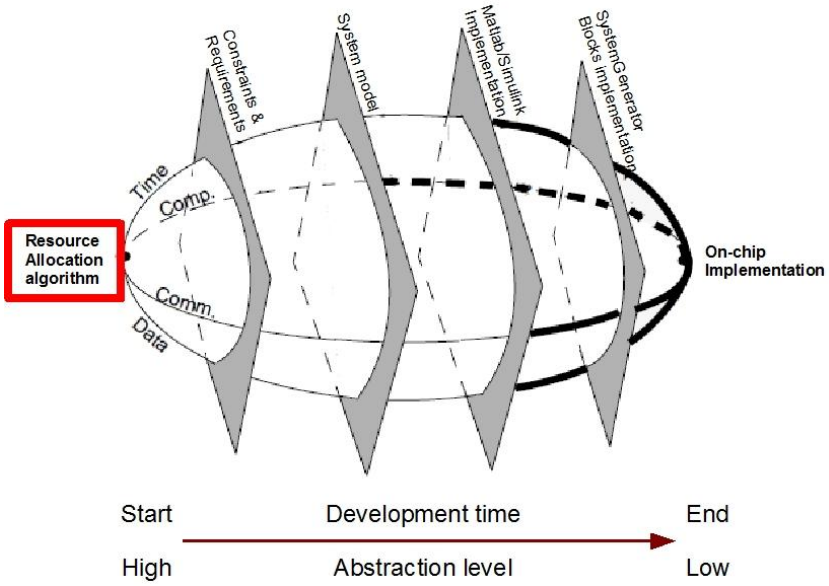

FIGURE 9 – MAPPING OF THE PROJECT DEVELOPMENT ON THE RUGBY META-MODEL

THE FIRST STEP OF THE PROJECT DEVELOPMENT IS THE DEFINITION OF THE STARTING IDEA.

The scope of the project is to analyze the issues related to the physical implementation of a dynamic resource allocation algorithm; this is the starting idea in relation to the chosen system design methodology presented in 1.5 (Figure 9).

In the following section of the report the description of the research on a dynamic allocation algorithm that constitutes the basis of this project implementation is presented. The algorithm that has been analyzed in this project is an implementation of the Spectrum Load Balancing (SLB) algorithm presented in [2], despite the main characteristics are maintained, the project considers a different operational environment.

The research on next generation (4G) communication technologies, such as International Mobile Telecommunications-Advanced (IMT-A), faces several issues related to the deployment of broadband connections in more dynamic environments compared to the past. The desired flexibility of networks where many operators are expected to operate at the same time on a shared channel, emphasizes the role of the resource allocation as a critic system design issue.
In this project the SLB algorithm operational context, is adapted in order to operate in the WiMAX simulation environment. In the following section a description of the algorithm is provided considering its original context. The detailed description of the implemented functionalities and the required assumptions is given in the next chapter.

1.9.1 FSU

In [1] the authors illustrate the principles that lead research on a new non-contention based method for spectrum sharing among different operators. The IMT-A systems are expected to significantly increase the network data-rates especially in Local Area (LA): such demanding achievements require the usage of MIMO antennas to improve the spectral efficiency and a wide spectrum allocation in the range of 100Mhz.
The allocation of such channel bandwidths leads to new technologic issues in terms of spectrum sharing among the operators, indeed it is considered not feasible to assign this amount of bandwidth to several operators in the same geographical area, as it is currently agreed for standards like Global System for Mobile communications (GSM) and Universal Mobile Telecommunication System (UMTS). Starting from this assumption, it becomes relevant to individuate new methods for channel sharing that allow different operators to coexist using the same frequencies and without interfering each others. The development of the Flexible Spectrum Usage (FSU) is meant to achieve this goal managing resource allocation in a joint frequency-time-space domain.

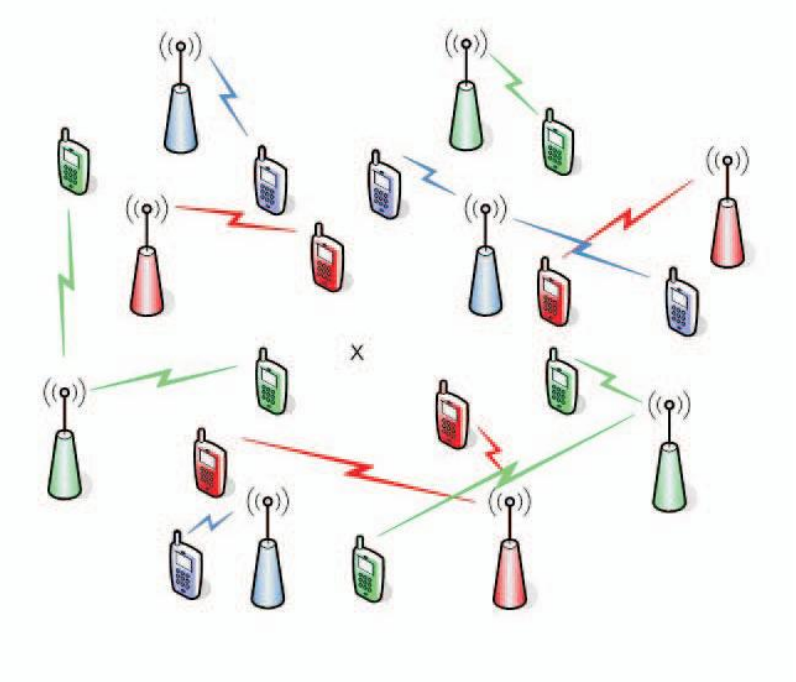

FIGURE 10 – EXAMPLE OF MULTI-OPERATOR ENVIRONMENT
DIFFERENT OPERATORS SHARE THE SAME CHANNEL IN A COMMON AREA. AN EFFECTIVE RESOURCE ALLOCATION IS REQUIRED [1].

FSU better works with OFDM systems because of their flexibility in spectrum allocation: the possibility of easily modify the transmitting carriers pattern is clearly a powerful tool in the resource sharing decision-making. The research focuses mainly on the Physical (PHY) layer of the system. The PHY layer has to be modified in order to implement the key features provided by FSU: spectrum sensing, interference avoidance and flexible spectrum allocation.
The analysis of Spectrum Control Strategies that enable the FSU implementation leads to the "Decentralized Approach" concept. In this approach a distributed spectrum allocation decision is considered as a more realistic solution for future large scale deployment of transmitting stations.
Another important concept related with the FSU implementation is the policy that manages resources utilization among the devices: in order to achieve good results a set of rules that promote a fair and efficient resources utilization has to be defined.

The Spectrum Load Balancing (SLB) algorithm, presented in [2], enables the FSU; it ensures the coexistence of several transmitting nodes (Home e-Node Base Stations - HeNBs) in the same area, preventing the mutual interference on the spectrum.
Basically the algorithm is developed following this two-steps concept: first a channel sensing is performed in order to determine the available resources and then a fair

allocation is executed among the active operators; in a second time the allocation process is improved by evaluating SINR thresholds on every Physical Resource Block (PRB). PRB is defined as the smallest frequency-time domain unit for spectrum allocation.

Here is a brief description of the algorithm as defined in [2] and illustrated in Figure 11.

1) Initialization Phase

Once it is activated, each HeNB allocates blindly the spectrum as per its requirement, before executing the algorithm. In this phase no channel information is available to the HeNB.

2) Spectrum allocation Phase

Step 1: Allocation of Free Spectrum

The algorithm first identifies the free spectrum in terms of PRBs. Are considered "free" those PRBs which have not been allocated by any HeNB. Free PRBs typically show no interference.
By using a Water Filling scheme (WF) the free spectrum is then allocated among the operators performing the algorithm in sequence on each HeNB. The sequence is known as part of the FSU policy. The following steps are then performed:

a) Each HeNB calculates the number of required PRBs according to a even distribution among all the operators; such average is obtained by dividing the sum of total number of allocated PRBs by each HeNBs plus the total number of free PRBs by the number of active HeNBs.

$$\frac{\sum_{n}^{NBs} A_{PRBs\ n} + N_{FPRBs}}{NBs} \qquad (1.9.1.1)$$

Where NBs is the number of the considered active HeNBs.

b) If the allocation of any HeNB is above the required average then the HeNB is excluded and a new threshold calculated.
c) Points a and b are repeated until all HeNB allocation satisfy the required average.
d) The spectrum is finally allocated among the selected HeNBs.

Step 2: SINR Based Spectrum Allocation

After the allocation of the free spectrum, the remaining amount of spectrum is selected considering a FSU target SINR threshold. The value of the threshold is specified in order to allow the coexistence of mutually interfering HeNBs. The PRBs above the threshold are selected and the HeNB will allocate only the required amount of PRBs. Step 2 ends if the HeNBs select the required amount of PRBs ore there are no PRBs left above the SINR threshold.

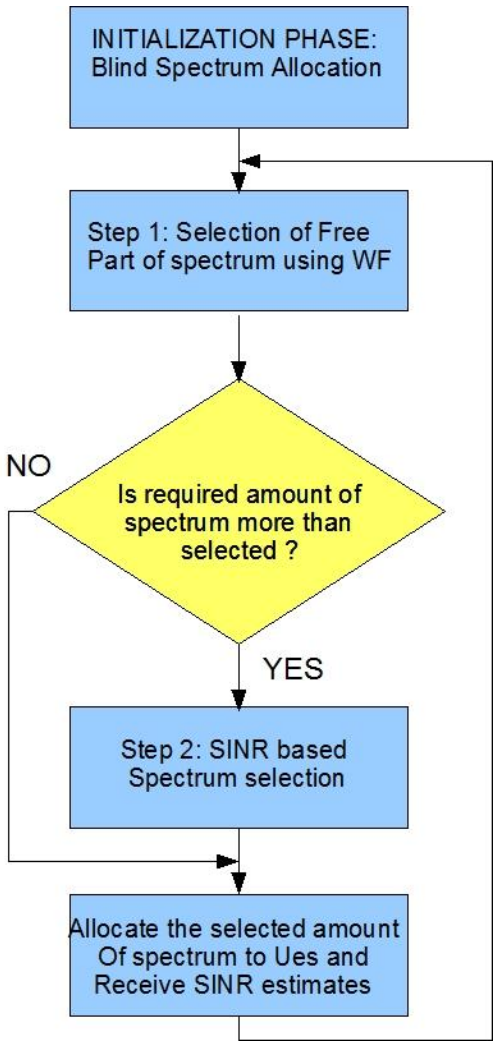

FIGURE 11 – FLOW CHART DESCRIBING THE ALGORITHM STEPS

1.10 System Definition

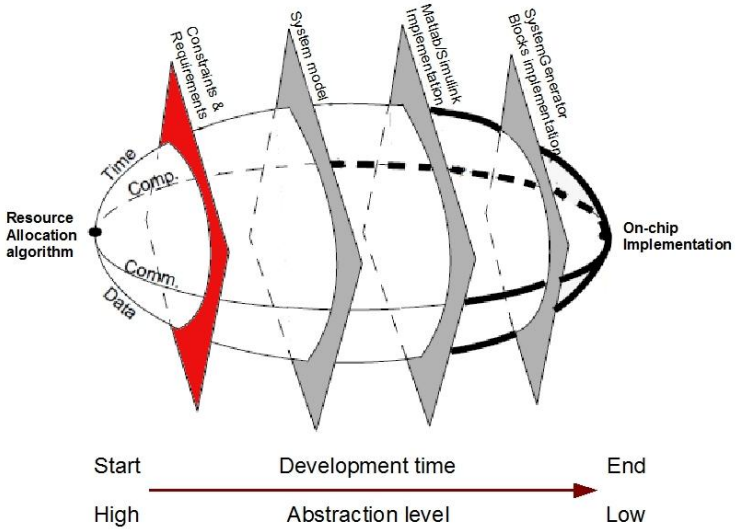

Figure 12 – Constraints and Requirements definition

In order to develop the starting idea, an analysis of the constraints and the requirements of the system has to be performed according to the proposed Rugby methodology (Figure12). This section of the report defines the development context of the algorithm implementation.

The algorithm analyzed in the project has initially been meant to be utilized in Local Area (LA), in particular environments characterized by random deployment of transmitting stations where coordination is a major issue hard to achieve. However, the philosophy that underlies the algorithm is also very interesting to be investigated and to be exploited in other different scenarios.

The choice of the communication standard is not the priority of this project, but in order to satisfy the key assumptions of the algorithm, an OFDM-based transmitting system that allows subcarriers allocation needs to be defined. In this analysis, a 802.16-2004 WiMAX-based model is used.

The communication environment configuration, is defined by three transmitting stations communicating with their receivers, interfering on a shared channel (Figure 13). In this project the three transmitter-receiver systems are named System 1, System 2 and System 3. The purpose of the implemented resource allocation algorithm is to avoid the mutual interference between the three transmitting systems.

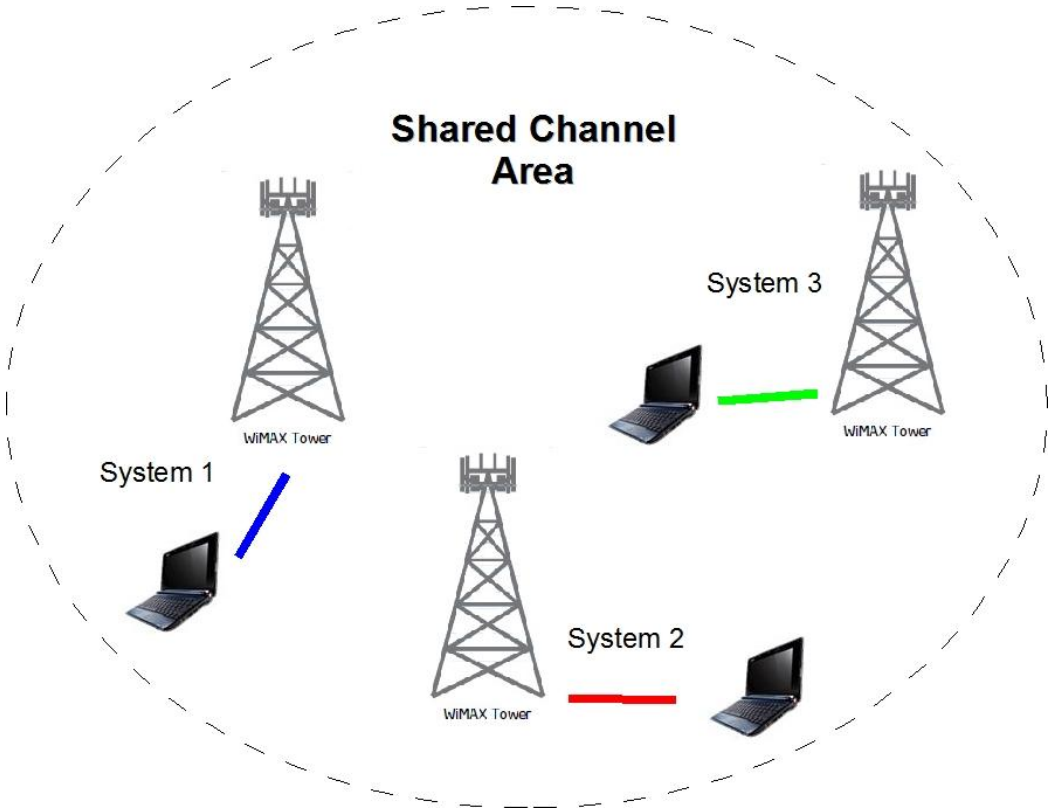

FIGURE 13 – SYSTEM CONFIGURATION

THE BASE STATIONS 1,2,3 COMMUNICATE WITH THE DEVICES INTERFERING EACH OTHER ON THE SHARED CHANNEL

Since the original algorithm research refers to IMT-A systems while this project implementation refers to WiMAX, confusion about the used terminology may occur. The main elements used in the description of the implemented algorithm functionalities are the same as the ones presented in 1.9.1 but the terminology is different.

From an algorithmic point of view, the operational functions of the Base Stations are the same as the HeNBs while the Subchannels constitutes the minimum allocable channel entities as the PRBs in the presented FSU algorithm. In the implementation description has been preferred to refer to Base Stations and to subchannels.

2. SYSTEM SIMULATION

2.1 LIMITATIONS AND ASSUMPTIONS

The implemented simulation environment only allows a fixed transmission configuration: the three Base stations start transmitting simultaneously and no random activation is allowed. This choice allows to reduce the system control required for the simulations.
The number of transmitting stations is fixed at three, this decision is given by the channel model implementation (presented in 2.2.2). The fixed number of active stations in the simulated environment allows to perform some simplifications at the System Generator level, reducing the complexity of the required control blocks.
The deployment configuration aims to have different interference power values among the three stations. The chosen pathloss values for the transmitted signals, are chosen in order to obtain an interference generation that suits the algorithm analysis and system behavior, rather than have real values for WiMAX environment. Table 2 shows the chosen pathloss values for the three systems.

	System 1	System 2	System 3
System 1	0	7dB	10dB
System 2	7dB	0	8dB
System 3	10dB	8dB	0

TABLE 2 – SIGNAL PATHLOSS AMONG THE THREE TRANSMITTING SYSTEMS

The starting OFDM transmission model and also the implemented system, works in downlink only: the channel allocation process is performed by the base stations and the chosen subchannels are allocated in downlink.
Following from the previous limitation, some assumptions in terms of BS-to-BS and SS-to-BS signalization are made: the algorithm requires to gather information about the allocation of the stations involved in the resource allocation process. This information is provided directly to the appropriate blocks, without considering transmission scheduling. Moreover, the receiver provides direct information about SNR and modulation rate to the transmitter.
The resource request of the base stations is also fixed: the multiple user management on a single station is not in the scope of this project. As previously explained in Section 1.7.2 of Chapter 1, the OFDMA is not enabled.
Different simulations with interesting schemes of resource allocation are provided. The goal is to test the algorithm behavior in different starting conditions where the channel allocation process is subjected to different levels of stress.

2.2 BASIC SIMULATOR

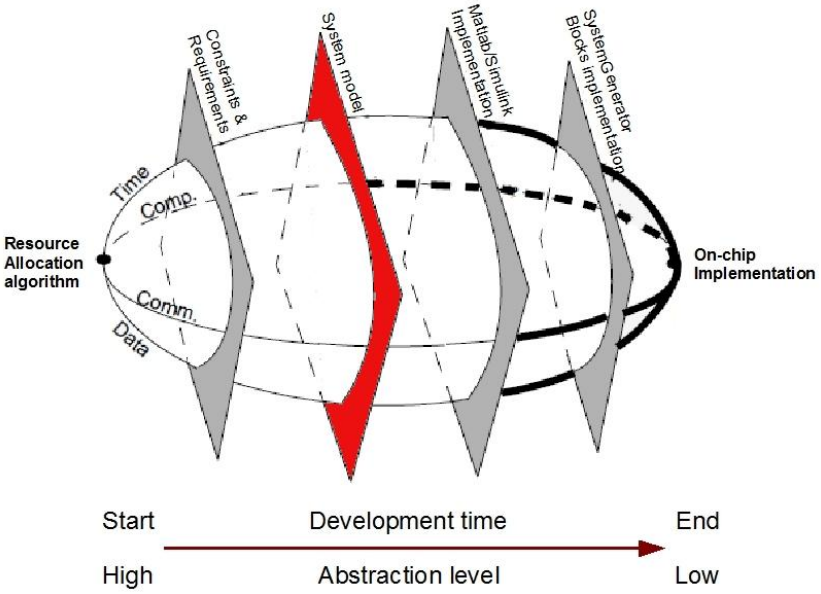

FIGURE 14 – THE SYSTEM MODEL ABSTRACTION LEVEL PROPOSED IN THE RUGBY META-MODEL AT SYSTEM MODEL THE NECESSARY ELEMENTS FOR THE IDEA IMPLEMENTATION ARE INDIVIDUATED

In order to implement the algorithm functionalities, it has been necessary first to individuate a proper simulation environment. The Mathworks Matlab/Simulink environment has been chosen as primary design and testing tool for the first part of the project.
Since the algorithm operates on a OFDM-based communication system, the first task was to implement the transmission environment on which the algorithm core has to be inserted.
The Matlab demo model "IEEE 802.16-2004 OFDM PHY Link, Including Space-Time Block Coding" [10] offers the basic functionalities required for developing the desired simulation environment. The Matlab demo constitutes the starting structure of the simulator that is then modified and extended in order to implement a more complex environment with 3 complete transmitter/receiver systems.

2.2.1 IEEE 802.16-2004 OFDM PHY LINK SIMULINK MODEL

The demo model is originally intended to experiment with the WiMAX profiles based on the IEEE 802.16-2004 standard defined by the WiMAX Forum. The demo implements the basic features of a 802.16-2004 based OFDM transmitter/receiver system: among the several tasks performed by the model, the following have been considered relevant for the project scope [10]:

- Generation of a random bit data that models a downlink burst consisting of an integer number of OFDM symbols.
- Adaptive modulation using one of the BPSK, QPSK, 16-QAM or 64-QAM constellations specified.
- OFDM transmission using 192 sub-carriers, 8 pilots, 256 points FFTs and variable cyclic prefix length.
- OFDM receiver including channel estimation.

In addition, the model uses an adaptive-rate control scheme based on Signal-to-noise (SNR) estimates at the receiver to vary the data rate dynamically, based on the channel conditions. The adaptive-rate control allows to evaluate the dynamic variation of the system throughput when the algorithm operates the resource allocation tasks.

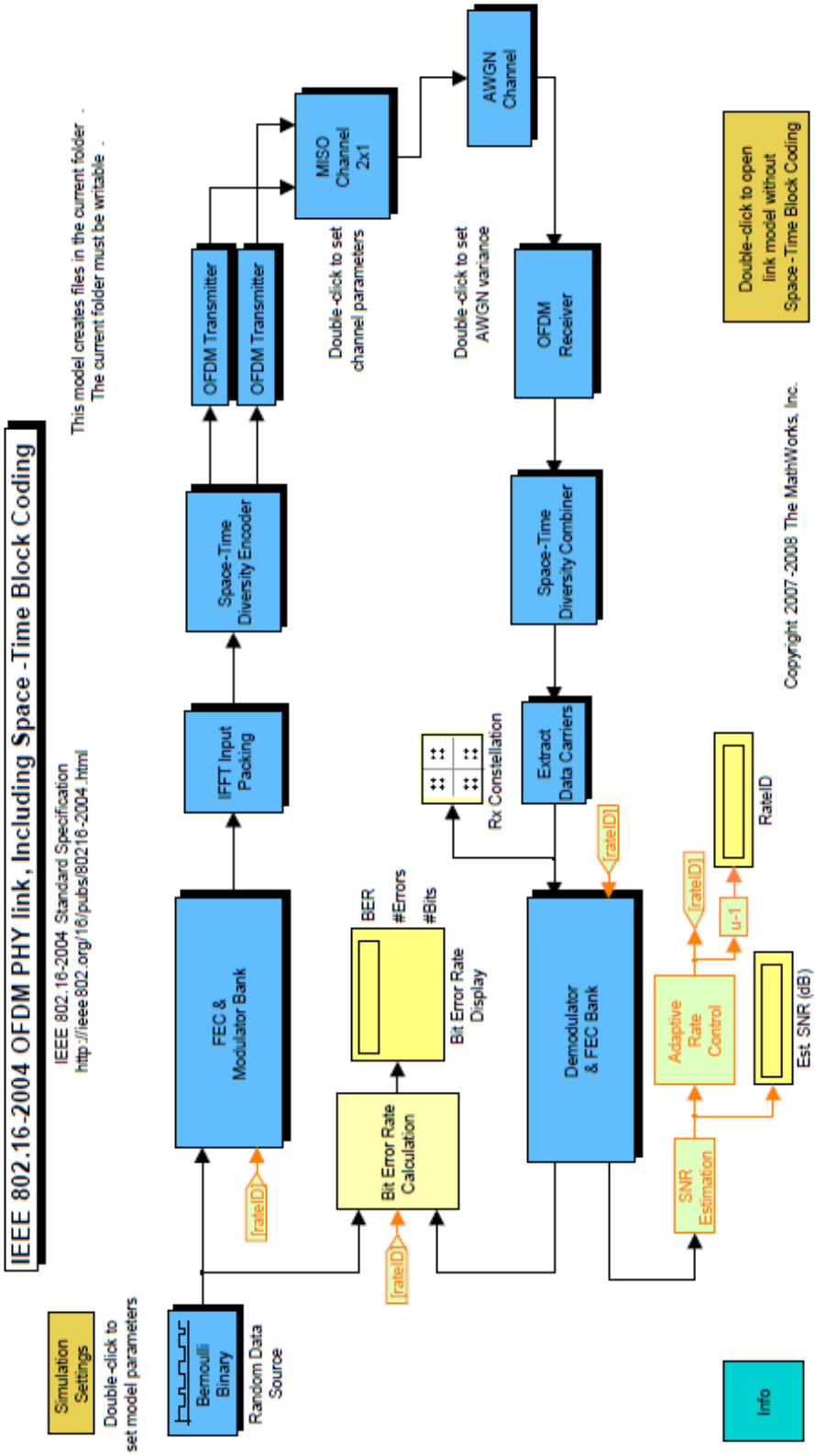

FIGURE 15 – THE STARTING SIMULINK DEMO MODEL[10]

Figure 15 shows the scheme of the IEEE 802.16-2004 OFDM PHY link simulink model that constitutes the starting structure of the simulation environment implementation. In blue color are depicted the communication structure blocks while in yellow and orange the analysis blocks and control signals.
The blocks implementing a OFDM /MISO transmitter are placed in the upper part of the scheme and the receiver chain blocks are placed in the lower half. The display in the middle provides information on the BER, number of errors and number of bits processed. On the bottom are placed the blocks estimating the SNR and managing the adaptive rate control.
Furthermore, the model allows to set few simulation parameters: Channel Bandwidth, OFDM symbols per burst, Cyclic prefix factor and the SNR thresholds for the Adaptive Modulation control. In this project the following parameters are considered: a channel bandwidth of 3.5 MHz, 2 OFDM symbols per burst and 1/8 cyclic prefix factor.
Modifying the SNR thresholds affects the system sensitivity in the modulation control; narrower thresholds cause the system varying the modulation more often in presence of variations of the SNR.
The default thresholds provided by the model are:

Modulation RS-CC rate	Threshold SNR value
BPSK ½	<4
QPSK ½	4
QPSK ¾	10
16-QAM ½	12
16-QAM ¾	19
64-QAM 2/3	22
64-QAM ¾	28

TABLE 3 – ADAPTIVE MODULATION THRESHOLDS

Running the model enables an OFDM transmission on a MISO channel. Information about the transmission spectrum and signal constellation currently active is then displayed. Every burst transmits 2 OFDM symbols simultaneously.

The model provides the basic required features to implement an OFDM based transmission system; in order to define the proper environment for the algorithm

simulations, several modifications are required. The desired configuration should perform the following tasks:

- Managing the transmission of three systems on a common channel.
- Perform the required channel sensing operations.
- Interference computation.
- Decision making functionalities according to the algorithm specifications.
- Sub-carriers allocation.

The starting model is then modified to support these features.

2.2.2 CHANNEL DESIGN

The demo model implements a MISO transmission channel: the OFDM symbols are transmitted on two separate channels that simulate a two-transmitters setup. The channel is modeled by two blocks : the MISO channel block, containing basically two Rician Fading simulation blocks, and a gaussian white noise AWGN block that allows to modify the channel noise parameter. The first modification is to utilize a Raleigh Fading channel block to model a system with Non-Line-Of-Sight (NLOS) characteristics. In Figure 16 the scheme of the implemented blocks for modeling the channel is depicted; the presented blocks constitutes the MISO channel block for one channel path. In addition to the Rayleigh Fading channel blocks, another block allows to modify the Free Space Path Loss parameter according to the considered path characteristics.

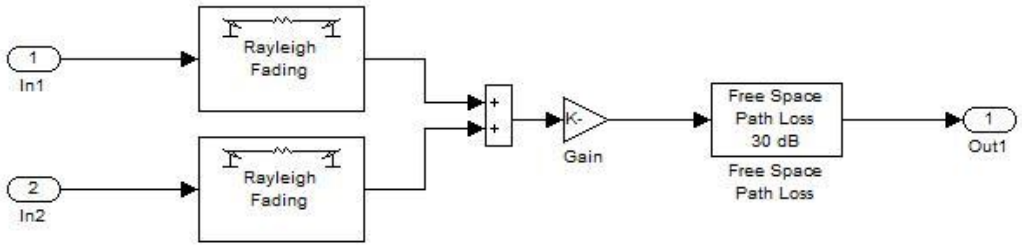

FIGURE 16 – THE IMPLEMENTED MISO CHANNEL BLOCK

In this project the shared channel is modeled considering all the different paths existing between the three systems: each system is composed of a transmitting base station and a receiver. The implemented solution provides the transmitting station with three connections to the three receivers and two more connections to the other transmitting base stations (Figure 17). The transmitter/receiver paths constitutes the feedback channels for the adaptive modulation control, as implemented in the original scheme

(Figure 15). The BS to BS paths are used for the channel sensing and interference computation purposes.

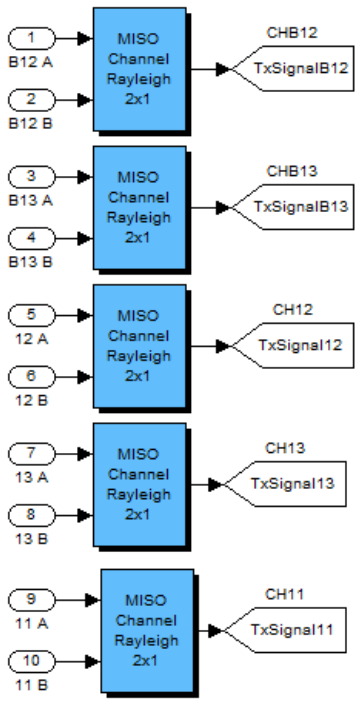

FIGURE 17 – CHANNEL PATH MODELS FOR SYSTEM 1

EACH TRANSMITTER USES 5 DIFFERENT CHANNEL PATHS TO MODEL THE SYSTEM CONNECTIONS.

The adequate paths are then summed in a single simulation signal to describe the channel behavior both at BS and at the receiver side. Figure 18 shows the channel implementation at the receiver.

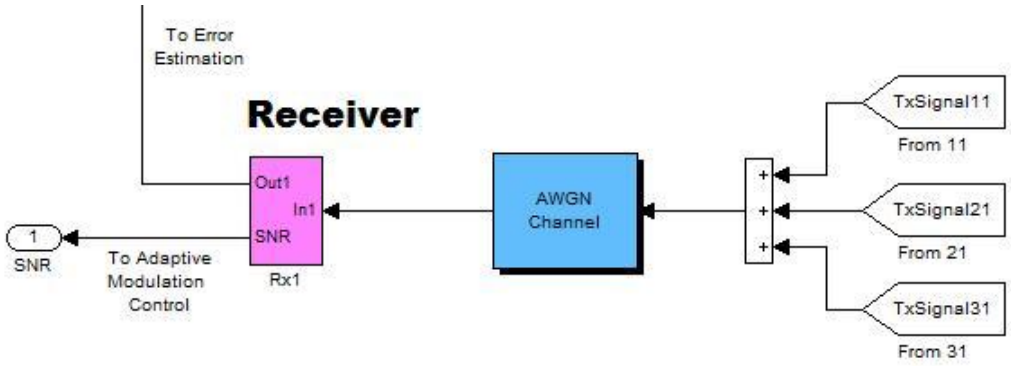

FIGURE 18 – COMPOSITION OF THE IMPLEMENTED CHANNEL PATHS AT THE RECEIVER SIDE

RECEIVER OF SYSTEM 1 SENSES THE CHANNEL CONTRIBUTIONS FROM BASE STATION 1, 2 AND 3.

The Receiver block demodulates the received signal and performs the SNR computation. In the original demo-model the received data stream are sent to the Error Estimation block that calculates the average error value and the total amount of transmitted bits.
Since the original block was designed for a single transmission usage, the error statistics and bit calculation features could not be used in this project implementation.

2.3 TRANSMISSION ENVIRONMENT

Given the previous channel setup, the effects of activating a simultaneous transmission on the three Base Stations are now considered.

In order to have reference SNR values for the project development, the first task is to simulate a transmission on free channel of every base station: every station allocates the full band available (48 subchannels) and transmits without interference.
In Figure 19 is shown the spectrum of the transmitted signal by System 2: all 48 subchannels are allocated.
In this case, System 2 transmits on a channel with a AWGN SNR parameter of 20dB, the average SNR at the receiver is 16.5 dB.

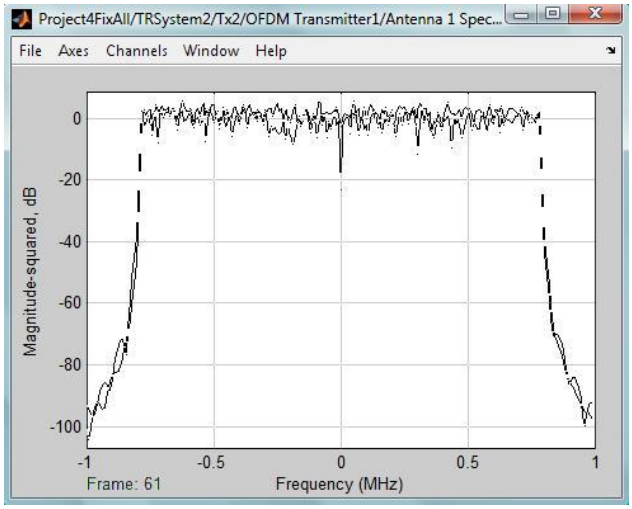

FIGURE 19 – SPECTRUM OF SYSTEM 2 TRANSMISSION

System 1 and System 3 (left and right side of Figure 20) have different channel paths: in the same noise conditions the average SNR at the receiver is 15dB and 19dB respectively.

If all the three station are activated and transmit with a fair allocation of 16 subchannels for each (Figure 20), the SNR values at the receiver remain almost equal to the ones obtained with a free transmission.

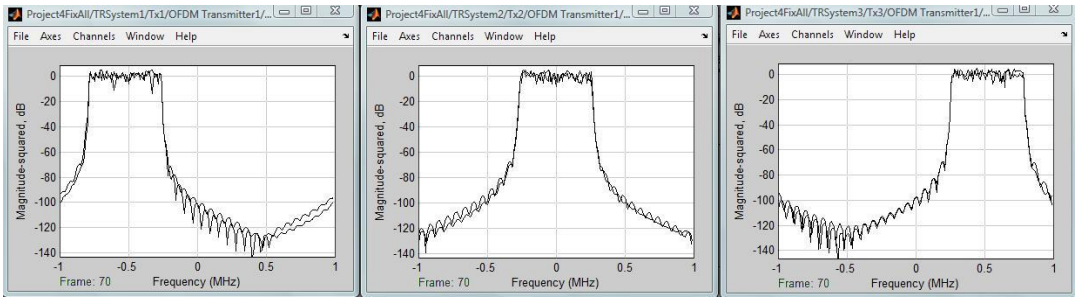

FIGURE 20 – ALLOCATED SPECTRUM OF SYSTEM 1, SYSTEM 2 AND SYSTEM 3 (16X16X16)

Another interesting example is to analyze the SNR values when the base stations transmit on the same subchannels generating interference. In Figure 21 the spectrum of the transmitted signals by System 1, 2, 3 is depicted: a different amount of allocated subchannels causes the transmitted signals to interfere.

In order to understand better the allocation context, in this example the subchannels are contiguously allocated: System 1 allocates 24 subchannels, System 2 allocates 32 subchannels and System 3, 16 subchannels. System 2 overlaps System 1 by 8 subchannels while System 3 is overlapped on all the 16 subchannels.

Subsequently System 1 has 16 not interfered subchannels, System 2 has 8, while System 3 none.

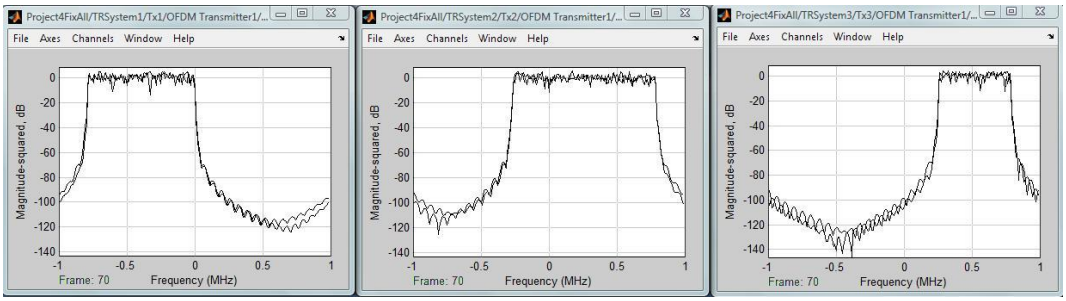

FIGURE 21 –TRANSMITTED SIGNAL SPECTRUM (24X32X16 ALLOCATION)

The SNR result values considerably vary in the three cases: System 1 maintains an acceptable average SNR of 12dB, System 2 decreases to 10dB while System 3 suffers a strong interference and lowers the SNR at the receiver to 3.5dB.

The purpose of the resource allocation process is to keep a fair allocation on the channel allowing the base stations to transmit simultaneously without interfering each other. Given the previous examples the goal is then to achieve SNR values close to the fair allocation example of Figure 20.

2.4 ADAPTED SLB ALGORITHM

The SLB algorithm presented in section 1.9.1 inspires the algorithm implemented in this project. The application context is different from the one individuated in [1] and [2], since the focus of the analysis is on the core algorithm functionalities rather than the communication standard. The choice of operating on a WiMAX environment does not affects the validity of the analysis since WiMAX is a OFDM-based system as required by the algorithm implementation.
In this section of the report, the implemented functionalities of the algorithm are presented.

The resource allocation algorithm is performed on all the stations in sequence and repeated at every allocation phase. The main steps are:

1. Initialization phase: random spectrum allocation at the activation of the station.
2. Allocation of the free subchannels.
3. Allocation of the occupied spectrum according to the sharing policy, resource requirements and interference threshold.

In the system design, 4 sub-carriers are considered as the minimum resource entity that can be allocated: 4 sub-carriers constitute a subchannel. A subchannel has the same characteristics than a PRB compared to the LSB original description.

2.4.1 TIME ORGANIZATION

The algorithm runs in sequence on every base station: a common time organization is assumed. The system operates the allocation and transmission processes in two different phases. In this project implementation 1/3 of the total time is dedicated to the channel sensing and resource allocation phase, while the remaining 2/3 are dedicated to the data transmission.

This choice clearly shows lack of efficiency since the total throughput can be raised reducing the resource-allocation-phase time, but it revealed as a good operative solution for the simulations. In figure 22 is depicted the time organization with different colors referred to the different transmitting systems. While System 1 (in blue) performs the channel sensing and the resource allocation running the algorithm, System 2 (red) and System 3 (yellow) transmit data. At the second system clock round, System 2 enters the sensing/allocation phase and System 1 and 3 transmit. Running this time schedule, a continuous resource allocation is performed by one station per time.

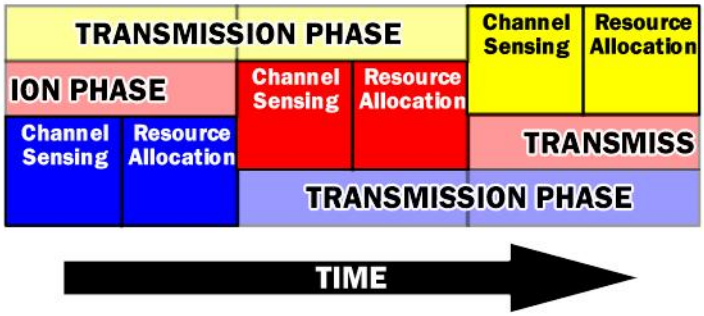

FIGURE 22 – GENERAL TIME ORGANIZATION.
SYSTEM 1 OPERATIONS IN BLUE, SYSTEM 2 IN RED, SYSTEM 3 IN YELLOW.

Table 4 shows the defined time values. The OFDM symbol time is given by the model specifications: every burst (transmission frame) two OFDM symbols are transmitted. The chosen clock period is equivalent to a 25 frames interval. The chosen settings enable a complete sensing/allocation phase every $7.2 * 10^{-3}$s.

OFDM symbol time (s)	$7.2 * 10^{-5}$
Defined System clock period (s)	$3.6 * 10^{-3}$

TABLE 4 - DEFINED SYSTEM TIME PARAMETERS

2.4.2 ALGORITHM OPERATIONAL STEPS

Figure 23 shows the control flow of the implemented algorithm.

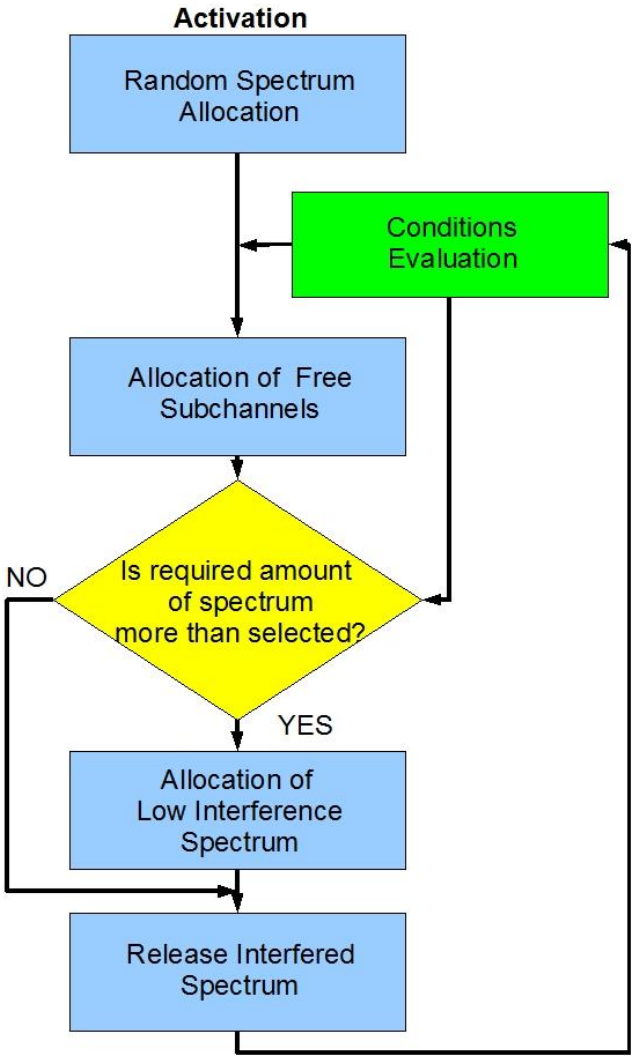

FIGURE 23 – IMPLEMENTED ALGORITHM PROCESS FLOW

At first, when the station is activated, a random spectrum allocation is performed. In relation to the number of subchannels needed, the station allocates randomly the required channel space. At this point it is assumed the interference of the system to be maximum since no information about the channel conditions is yet available.

After the first allocation, the information about the interference on the channel is available. The algorithm evaluates then how many subchannels are free and available to be allocated. The number of free subchannels that can be allocated by a single station is given by the computation process presented in (1.9.1) [2].

After the allocation of the free subchannels, the algorithm checks if more subchannels are needed. In case of more resources required, the algorithm selects the subchannels that

have a tolerable interference value (in the project a value of -7dB is used) and allocates the required number as long as there are subchannels available.

The last step is concerned with releasing the subchannels that were currently allocated but have a not tolerable level of interference. These subchannels will be available for the resource allocation of the following station on the schedule.

After the activation phase, the random allocation is no more executed: an evaluation of current allocation conditions, involving the number of subchannels allocated in relation to the number of subchannels required is performed instead.

2.5 REALIZATION DETAILS

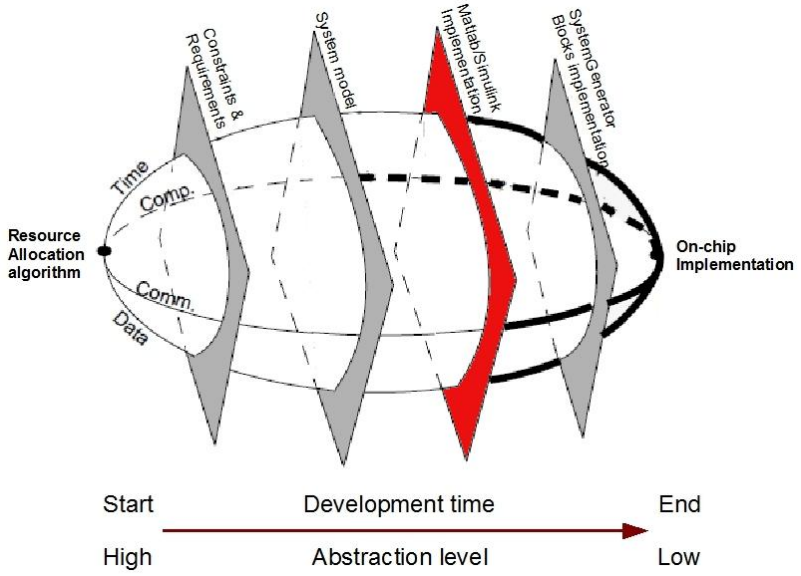

FIGURE 24 – THE MATLAB/SIMULINK IMPLEMENTATION LEVEL ON THE RUGBY META-MODEL

In this section of the report, the details about the implemented solutions in the Matlab/Simulink model are presented. Figure 24 provides the positioning on the Rugby meta-model abstraction levels scheme of the project development

2.5.1 CHANNEL SENSING

As previously mentioned, the algorithm needs the channel interference estimates in order to operate correctly. The implemented channel sensing blocks provide such information to the station that is performing the resource allocation phase. The scheme in Figure 25, shows the implemented blocks in the simulation model for System 2: the signals from the

stations 1 and 3 are added on the BS-to-BS channel and processed by the Interference Power Estimation block. The result of the estimation is a vector of 48 values, one for each subchannel, that is then given in input to the Decision Making block.

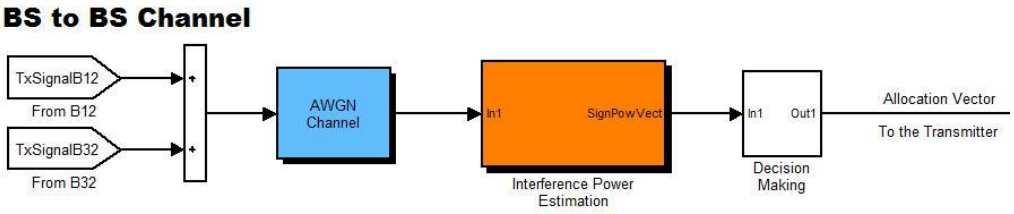

FIGURE 25 – THE INTERFERENCE ESTIMATION BLOCK CHAIN

Basically the Interference Power Estimation blocks extracts the subcarriers data from the data stream on the channel (the transmitted signal on the channel contains several data that are not used for computation; for example the cyclic prefix, pilots carriers, preamble, etc.) and computes the power of the signal. The signal evaluated is composed by interference given by the other Base Stations and noise. The power of the interference on a subchannel is given by the mean of the composing subcarriers power over the two transmitted frames as showed in (2.5.1.1).

$$P_{Sch} = \frac{\sum_{F1} P_{Scar} + \sum_{F2} P_{Scar}}{8} \qquad (2.5.1.1)$$

where P_{Sch} is the power on the subchannel, P_{Scar} the power on the single sub carrier, and F_1 and F_2 the two transmitted frames.

The decision of working directly with an interference power vector instead of the SINR ratio as proposed in [2] is a limitation of this project that reduces the simulation complexity. Using a SINR values vector however, does not imply significant modifications in the Decision Making block where the algorithm is executed.

2.5.2 DECISION MAKING

The developed Decision Making block, contains the core blocks where the algorithm functionalities are implemented. In Figure 26 a scheme of the blocks is presented: the interference power vector constitutes the input of the scheme, the output is the allocation mask vector that will be used for enabling the desired subcarriers.

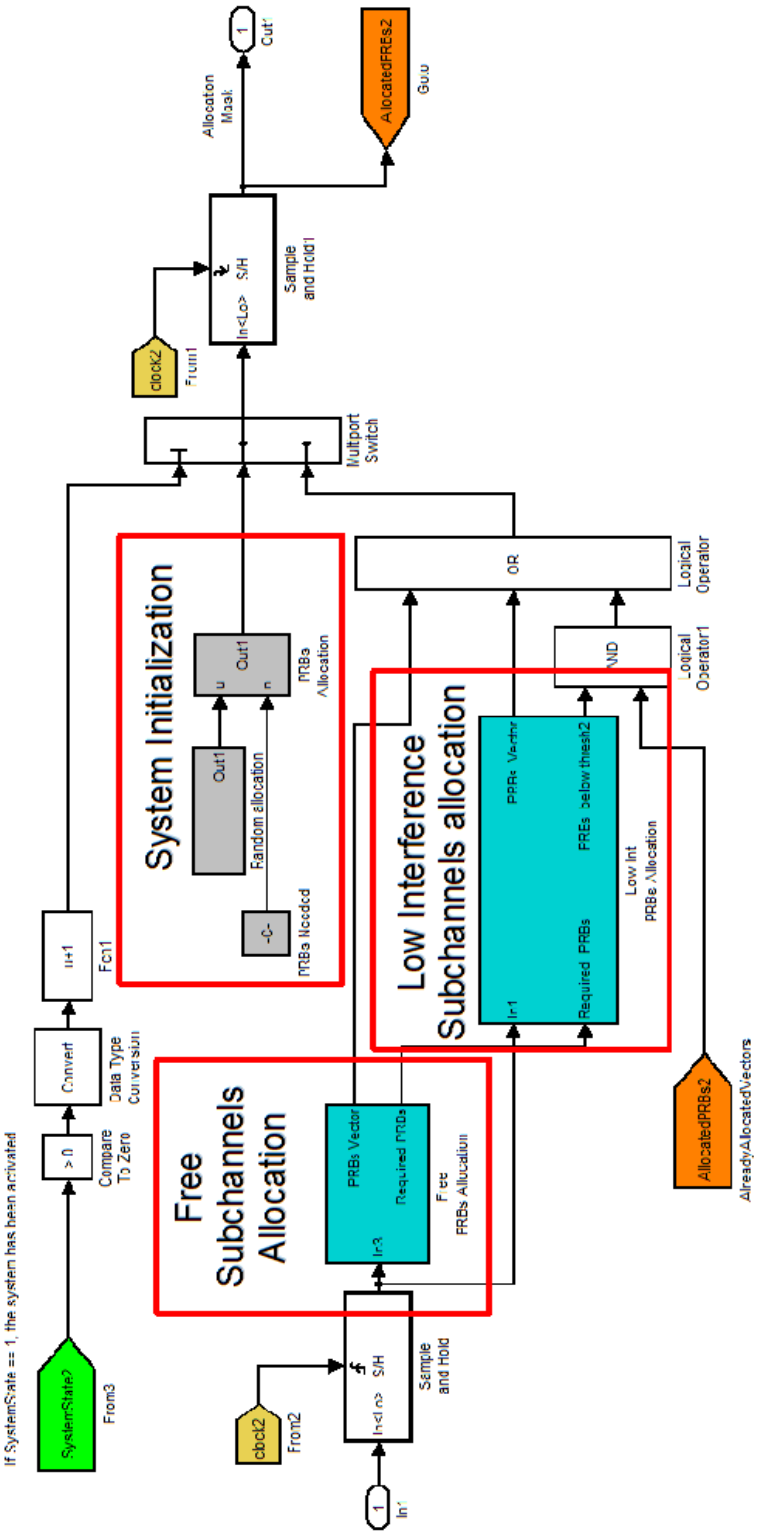

FIGURE 26 – DEVELOPED DECISION MAKING BLOCK SCHEME

The grey blocks are used for the initial random allocation phase while the cyan blocks implement the resource allocation phase functionalities. The sample and hold blocks and the clock signalizations are used for simulation purposes.
The Decision Making blocks performs the three main tasks of the algorithm:

- Initial random allocation.
- Allocation of free subchannels.
- Allocation of required low interference subchannels.

In the Matlab/Simulink simulation model all these tasks are performed with Simulink blocks and Embedded Matlab functions blocks. The use of embedded functions is a high abstraction level implementation that speeds up the design process and allows to evaluate the global behavior of the algorithm without concentrating on low-level details.
The embedded function blocks cannot be handled as such for hardware implementation level and require a later re-design process discussed in 3.2.

Both Free subchannels allocation and Low-Interference (LI) subchannels allocation blocks are organized in a three-steps process :

1. Sorting of the interference power vector (ascending order, low interference power values on top).
2. Threshold evaluation to individuate free or low-interference subchannels.
3. Creation of a “mask” vector where the valid (free or LI) subchannels have value “1” and not valid have “0”.

Figure 28 shows the inner organization of the developed Free PRBs Allocation Block. At first the interference power vector is sorted with a quicksort algorithm; the yellow block then performs the comparation with the threshold and outputs a vector of 48 elements (each per subchannel) where free subchannels are marked with “1” (Figure 27).

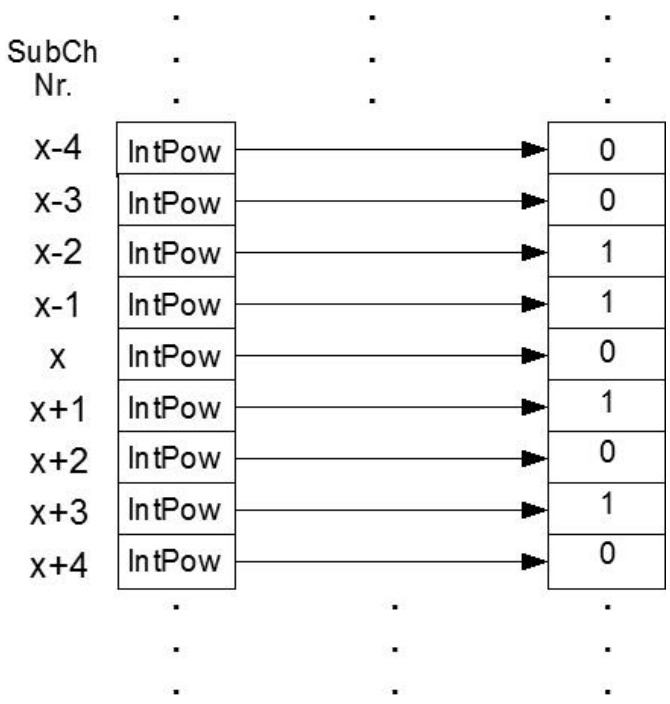

FIGURE 27 – EXAMPLE OF "MASK" VECTOR

THE VECTOR WITH INTERFERENCE POWER IS USED TO CREATE A BINARY MASK VECTOR WHERE SUBCHANNELS SATISFYING THE THRESHOLDS VALUES ARE MARKED WITH "1".

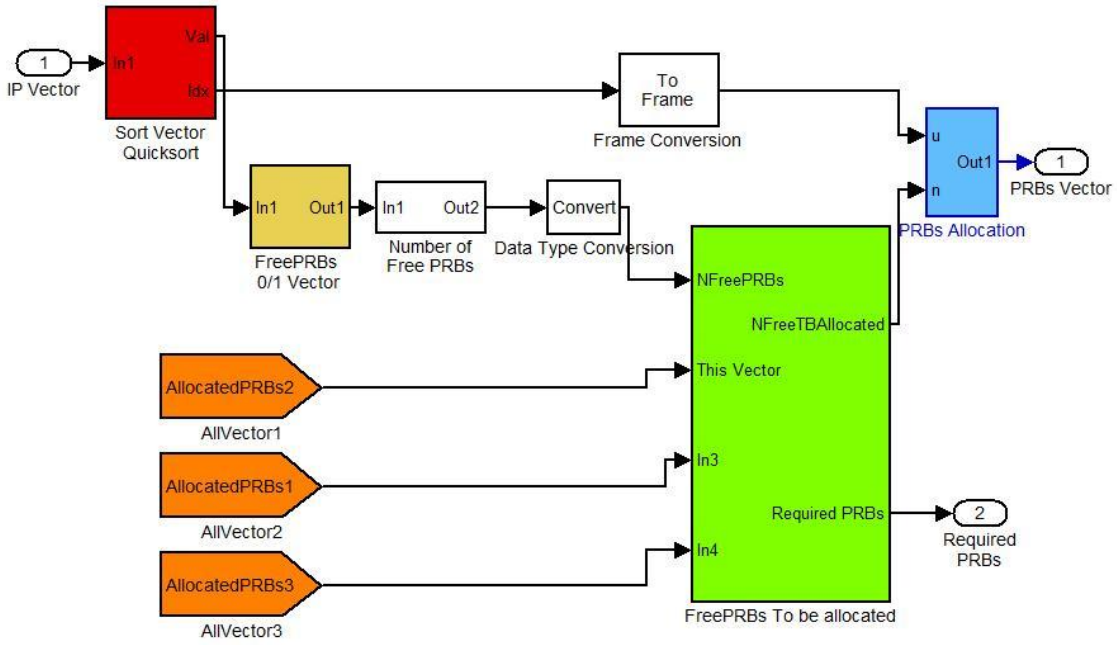

FIGURE 28 – FREE PRBS ALLOCATION BLOCK

The orange tag blocks simulate the signalization between the stations which have to communicate how many subchannels they have allocated for own transmission, as proposed in [2].

The green block is showed in Figure 29.

In order to calculate how many free subchannels the station is allowed to allocate, at first is computed how many subchannels are already allocated and then the fair allocation process presented in section 1.9.1 is performed. The fair allocation process is one of the core functionalities of the implemented algorithm and at this level, is implemented with an Embedded Matlab Function block.

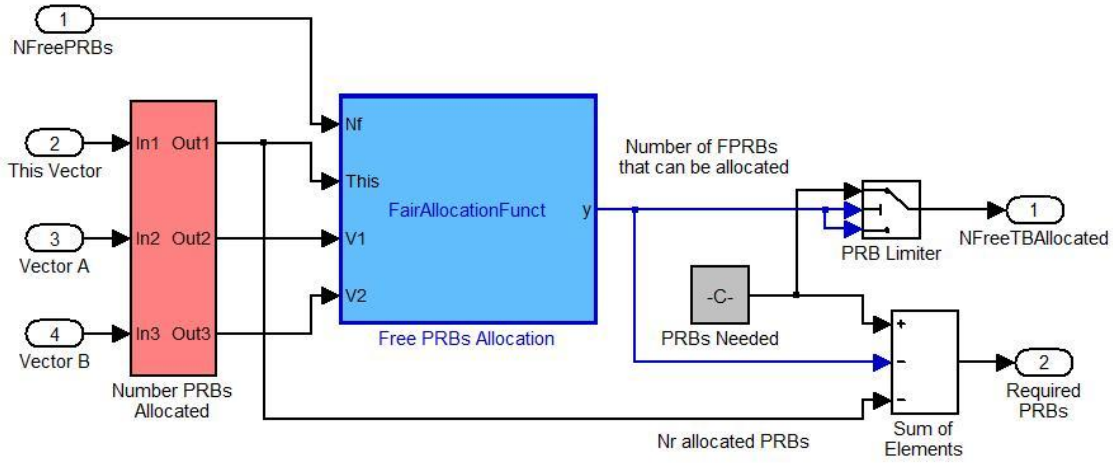

FIGURE 29 – FREE PRBS TO BE ALLOCATED BLOCK

As shown in Figure 29, the total number of free subchannels that can be allocated is given by the number of available subchannels, limited by the number of limited subchannels. In this project we assume a constant resource requirement thus this value is given by a constant initialized by the simulation.

Finally the number of required subchannels is given in output to be processed by the LI allocation blocks.

The allocation of LI subchannels is performed in the Low Interference Allocation block magnified in Figure 30.

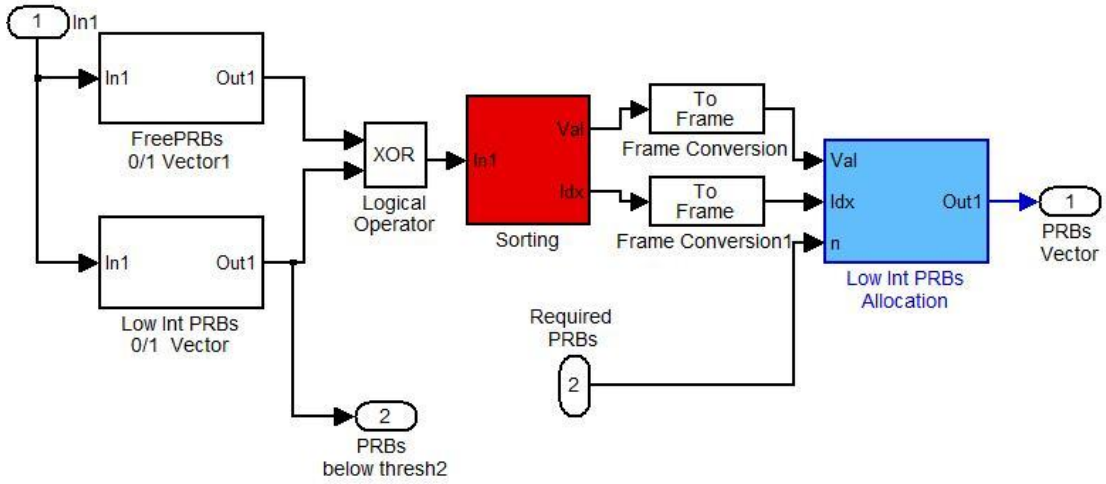

FIGURE 30 – LI PRBS ALLOCATION

The first two blocks on the left perform a comparison with the thresholds in order to individuate the low interference subchannels. The information about the number of required subchannels is then utilized to create the allocation mask vector.

The final allocation vector that the Decision Making block outputs to the transmitter for subcarrier selection, is composed by three contributions : the Free subchannels vector, the Low Interference subchannels vector, and the previous allocation vector updated in relation to Interference values.

2.6 COMPLETE SIMULATION ENVIRONMENT

The simulation environment is completed by inserting the channel sensing block and the Decision Making block, into the original 802.16-2004 OFDM transmission system.
The output of the Decision Making block is a vector of binary values that are then utilized to enable the subcarriers in the IFFT Input Packing block of the OFDM transmission model.
In Figure 31 the IFFT Input Packing block scheme is depicted. The data stream coming from input 1 are reshaped in order to simulate two frames of 192 (number of subcarriers) elements. The allocation vector is given in input 2 and is composed by 48 elements. The Subchannels Enabling block is an Embedded Matlab function that multiplies the 192-dimensional vector with the allocation vector : in order to have 48 subchannels from the starting 192 subcarriers group, 4 elements of the data frame vector are multiplied by the same allocation vector value. If the subchannel is not to be allocated, the data are

multiplied by zero and the considered group of 4 subcarriers does not transmit over the channel.

The choice of introducing this kind of subcarrier suppression is obviously not feasible in a real system because part of the data are lost and not transmitted. The implementation of a complete data management related to the utilized subcarriers is not in the scope of this project: it is, however, a mandatory feature for real-system applications.

The IFFT Input Packing Block performs the pilot insertion as implemented in the original design. Since the subcarrier suppression is introduced, a pilot suppression is also implemented in order not to have pilots in unused part of the spectrum.
The original Matlab OFDM transmission model does not utilize the introduced pilots: even in the modified model implemented in this project the pilot management for channel estimation is not considered.

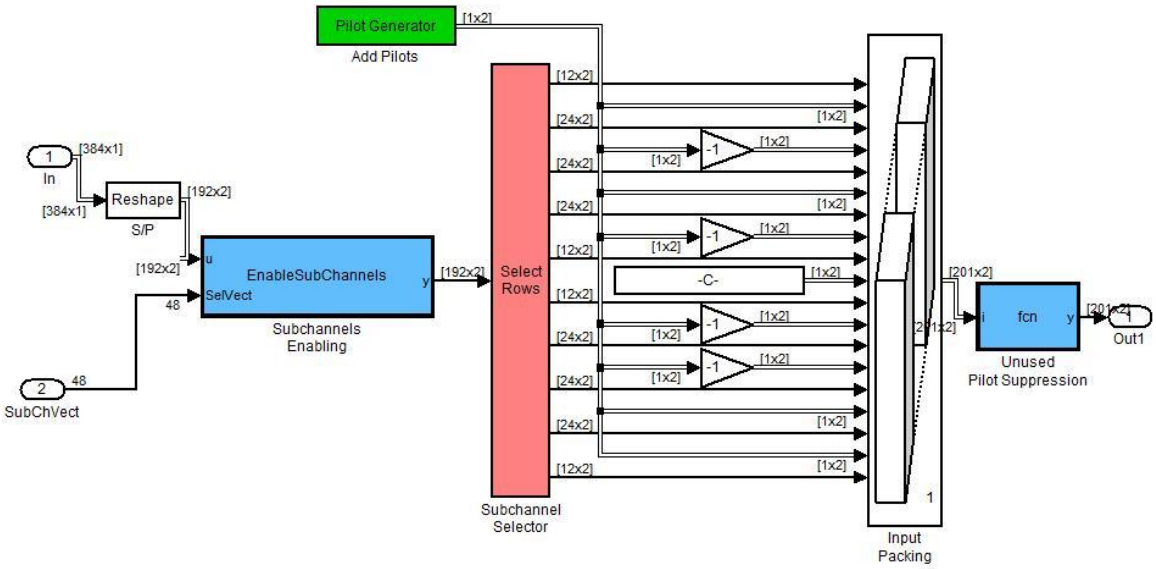

FIGURE 31 – THE IFFT INPUT PACKING BLOCK

THE DIMENSIONS OF THE SIGNALS ARE REPORTED IN THE BRACKETS.

In order to validate the functioning of the implemented system model, various simulation are performed in different starting conditions. The results obtained from the simulations, are presented in Chapter 4, Section 4.2.

3. HARDWARE IMPLEMENTATION

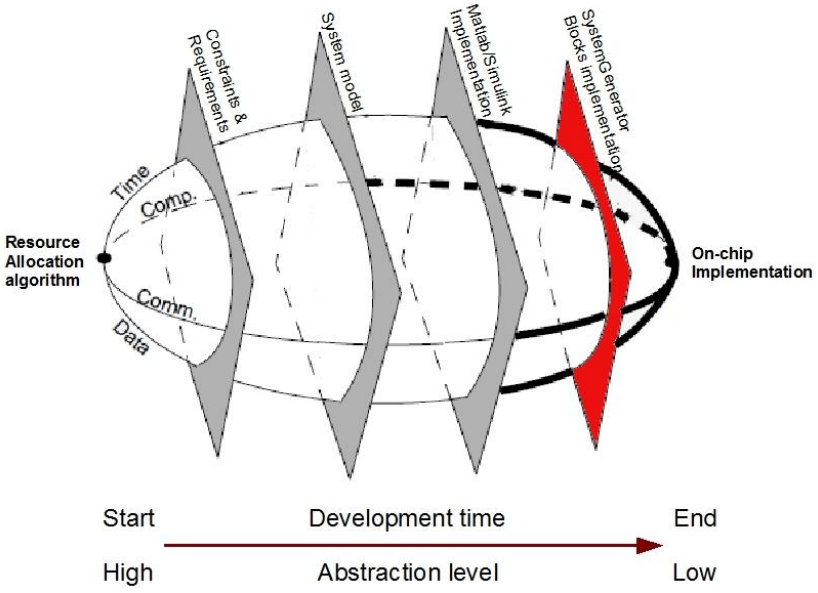

FIGURE 32 – THE SYSTEM GENERATOR BLOCKS IMPLEMENTATION ABSTRACTION LEVEL MAPPED ON THE RUGBY META-MODEL

After the implementation of the desired algorithm functionalities in the simulation environment, it is very interesting to understand what are the issues and the constraints related to a hardware implementation of the system.
The feasibility of a system implementation on an FPGA board constitutes a clear element of interest for scientific and commercial purposes.

In this chapter of the report, an analysis of the issues related to the FPGA design of the algorithm functionalities is provided.

3.1 XILINX SYSTEM GENERATOR

System Generator is a design tool that enables the use of Matlab/Simulink model design environment, for Xilinx FPGA design [17]. A previous knowledge of FPGAs or RTL (Register Transfer Language) design methodologies is not required for system design, since System Generator provides a specific blockset that allow FPGA design in the friendly Simulink environment.

Figure 33 shows how the design flow allows to move from a Matlab/Simulink implementation of the system, to the FPGA implementation.

Once the functionalities and the basic data-flow issues have been defined, System Generator is used to specify the hardware implementation details for Xilinx devices.

At first, the system design is performed by using the provided Xilinx DSP blockset for Simulink. System Generator generates RTL automatically invoking the Xilinx Core Generator. After this the RTL is synthesized to obtain an optimized netlist for the used blocks together with map, place and route functions.

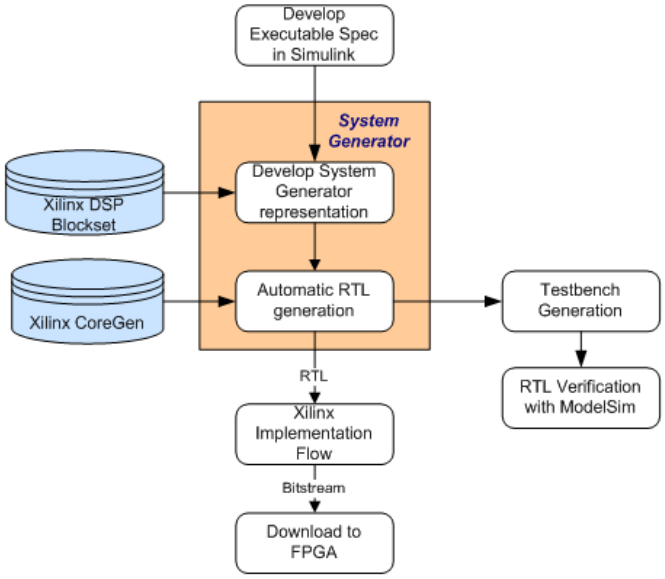

FIGURE 33 – SYSTEM GENERATOR DESIGN FLOW [17]

The bistream required for FPGA programming is finally generated by System Generator, by executing the downstream implementation tools.

In this project the high level implementation of the algorithm functionalities is performed in Matlab/Simulink environment, the target platform is a Xilinx ML506 Evaluation Platform [12]: System Generator is then considered as an “optimal” tool to perform the FPGA design of the desired functionalities.

3.1.1 XILINX DSP BLOCKSET

The provided Xilinx DSP blockset is a block library that contains more than 90 building blocks for FPGA design. It is accessible by the Simulink Library and allows to utilize the provided blocks as regular Simulink blocks.
The building blocks provide the functionalities of different devices that can be implemented on FPGA. The complexity level can vary from basic elements as logic ports, inverters, multiplexers, to more complex elements as RAM memories, dividers or communication-oriented devices as encoders, channel models, interleavers.
Each block allows to manage the featured parameters and to manage the output signal properties. When dealing with Xilinx System Generator design, the analysis of input/output signal properties is always a critical issue. System Generator operates with fixed-point numbers while Matlab/Simulink works with floating-point values. In the implemented blocks the used input values are integer numbers that require up to 6 precision bits (6 bits allow the representation of 64 non signed values).

3.2 SYSTEM DESIGN

The core of the algorithm functionalities is represented by the system blocks performing the fair subchannels allocation descript in (2.5.2). In this project the Free PRBs Allocation block, containing all the relevant algorithm features, is implemented with System Generator.
As depicted in Figure 34, the Free PRBs block model is mainly defined by three functions:

1. Sorting block
2. Vector creation
3. Fair allocation computation.

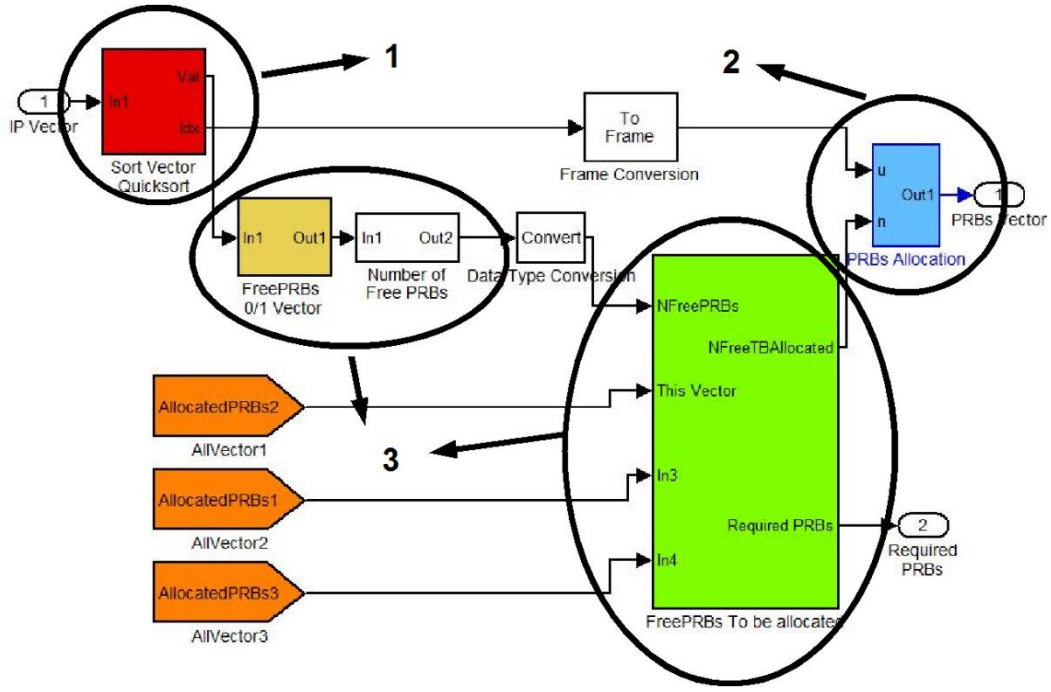

FIGURE 34 – ANALYSIS OF THE FREE PRBS BLOCK

The hardware implementation of a sorting algorithm is a important issue since the requirements in terms of physical resources and computation time are not neglegible especially when the algorithm needs to perform several sorting operations.
However, since the design of the sorting block is matter of extensive research and requires a relevant project time, the priority has been given to the implementation of the algorithm specific functionalities individuated in 2 and 3 in Figure 25. The sorting block features are realized with Simulink blocks.

At this point of the project, the Xilinx System Generator version of the developed design (Section 3.3) has been tested as a standalone version only. Due to time limitation it has not been tested with the rest of the Simulink model chain. However, it is expected that the complete chain including the Xilinx System Generator block will be tested once the report has been handed out, and that the results will be presented during the examination.

3.3 IMPLEMENTED ELEMENTS DESCRIPTION

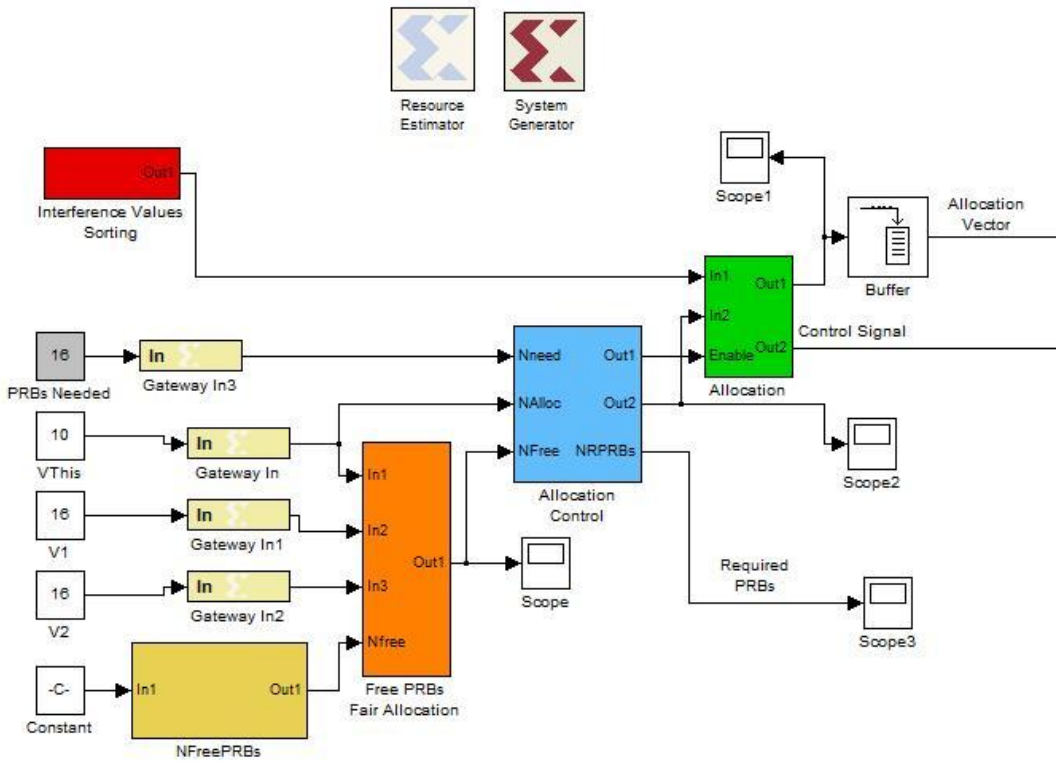

FIGURE 35 – THE IMPLEMENTED FREE ALLOCATION BLOCK DESIGN MODEL

In Figure 35 the general scheme of the implemented block is depicted.
On the top of the model are showed the two System Generator tokens required to run the simulation, to perform the model low-level code compilation and to run the resource estimation.

3.3.1 IMPLEMENTED ELEMENTS

Figure 36 shows the available options for the System Generator token: the desired target platform is specified for the compilation. In the lower part of the window are showed the clock settings in order to correctly operate in the Simulink environment.

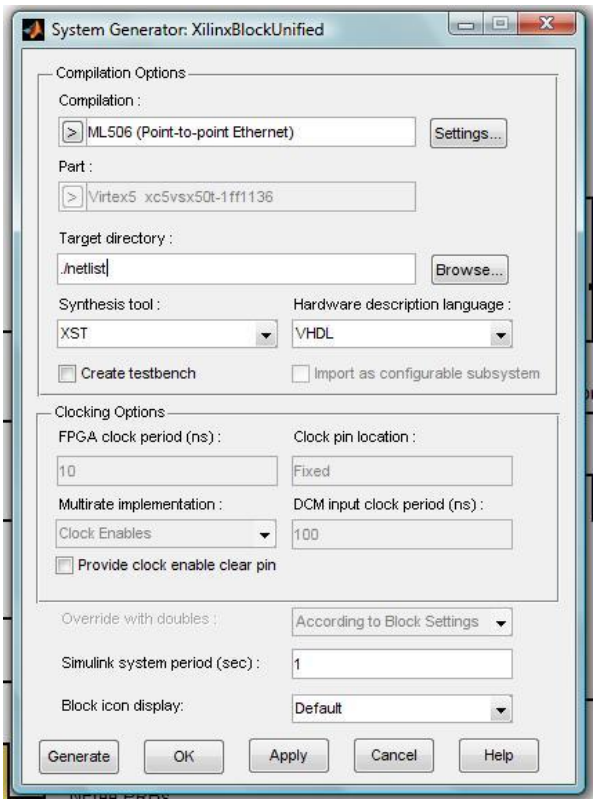

FIGURE 36 – OPTIONS PROVIDED BY THE SYSTEM GENERATOR TOKEN

In the model the Gateway input/output ports (Figure 37) are required for fixed-point numbers conversion: the ports also specify the output precision in terms of bits and signing method.

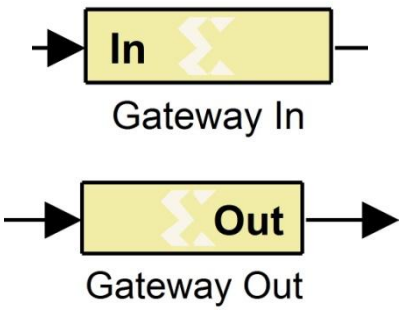

FIGURE 37 – THE GATEWAY IN/OUT PORTS

Typically the Gateway ports constitutes the boundaries of the FPGA system that can then be inserted in a Simulink environment. Inside the boundaries all the input ports of the used System Generator blocks, must be driven by other Xilinx blocks, no Matlab/Simulink blocks are allowed. In Figure 38 a scheme of a combined Matlab/Simulink-Xilinx System Generator model is depicted.

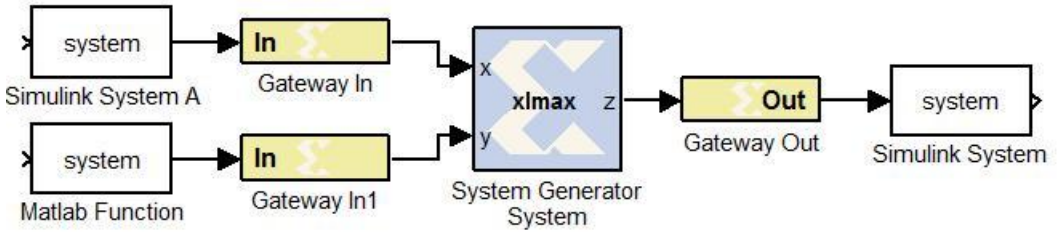

FIGURE 38 – COMBINED SIMULINK/SYSTEM GENERATOR MODEL SCHEME
BETWEEN THE GATEWAY PORTS ONLY SYSTEM GENERATOR BLOCKS ARE ALLOWED

In the implemented stand-alone setup, five inputs are provided:

1. The sorted interference power vector
2. The number of needed subchannels
3. The number of allocated subchannels by the running station
4. The number of allocated subchannels by station 2
5. The number of allocated subchannels by station 3

In Figure 35 scheme an additional input is provided for the computation of the free subchannels: this operation can be performed using the interference power vector as input but in order to have greater testing flexibility it has not been implemented.

The first operation is the computation of the number of free subchannels available, this is performed by the NFree PRBs block (yellow in Figure 35). The single elements of the interference power vector are compared to a fixed threshold that defines a free subchannel: the comparison is performed with a Relational block.
Figure 39 provides a detail of the NFree PRBs block: the 48 interference power signals are processed separately: in this implementation a Simulink demux is used to extract the signals from the original vector.

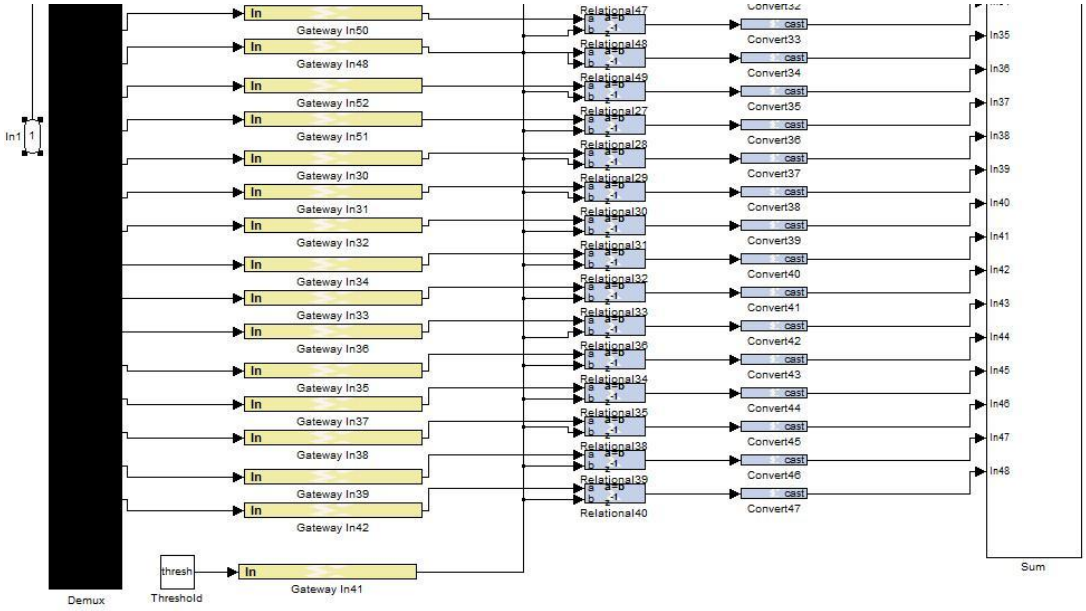

FIGURE 39 – DETAIL OF THE NFREE PRBS BLOCK

THE 48 ELEMENTS OF THE ARRAY ARE COMPARED TO THE THRESHOLD AND THEN SUMMED. THE TOTAL FREE SUBCHANNEL NUMBER IS IN OUTPUT.

Every Relational block (Figure 40) performs a comparison with a standard delay of z^{-1}, the output is a boolean value. If the condition is satisfied the output is true.

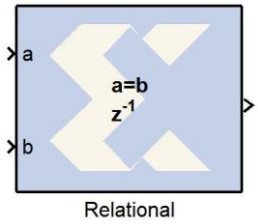

FIGURE 40 – RELATIONAL BLOCK PROVIDED BY THE SYSTEM GENERATOR BLOCKSET

The sum is performed by the implemented sum block.

The basic AddSub block provided by the System Generator blockset manages two inputs only, so the required 48-elements Sum block is implemented with a chain of AddSub blocks as depicted in Figure 41.

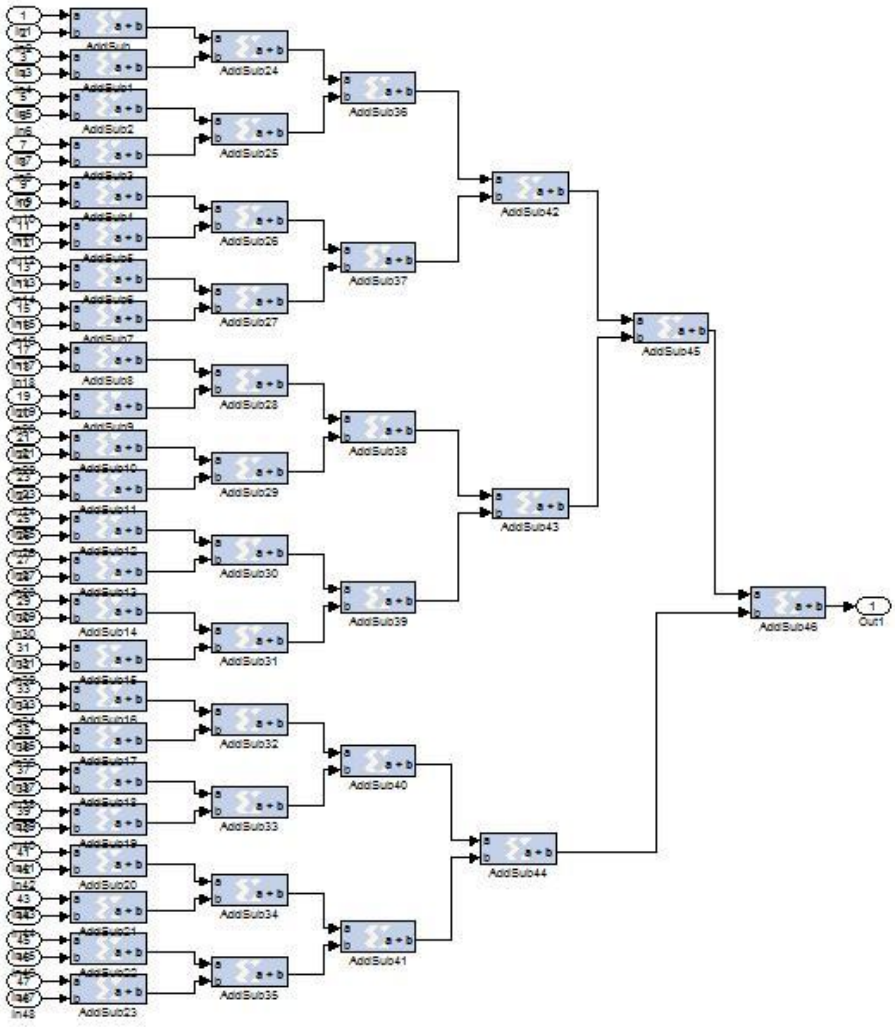

FIGURE 41 – IMPLEMENTED SUM BLOCK

As long as the AddSub blocks does not accept the boolean output of the Relational block, a conversion (Cast) block is required. The Cast block converts the boolean values in a binary unsigned value. In the implementation detail of Figure 39, the Cast blocks are used to adapt the output of the Relational blocks to the input of the Sum block.

3.3.2 FAIR ALLOCATION IMPLEMENTATION

The number of free subchannels is used as input for the following block that computes the number of free subchannels that can be allocated by the station. This block named Free PRBs Fair Allocation, performs one of the core functionalities of the algorithm implementing the fair allocation strategy presented in the previous chapters.
The block model scheme is presented in Figure 34.

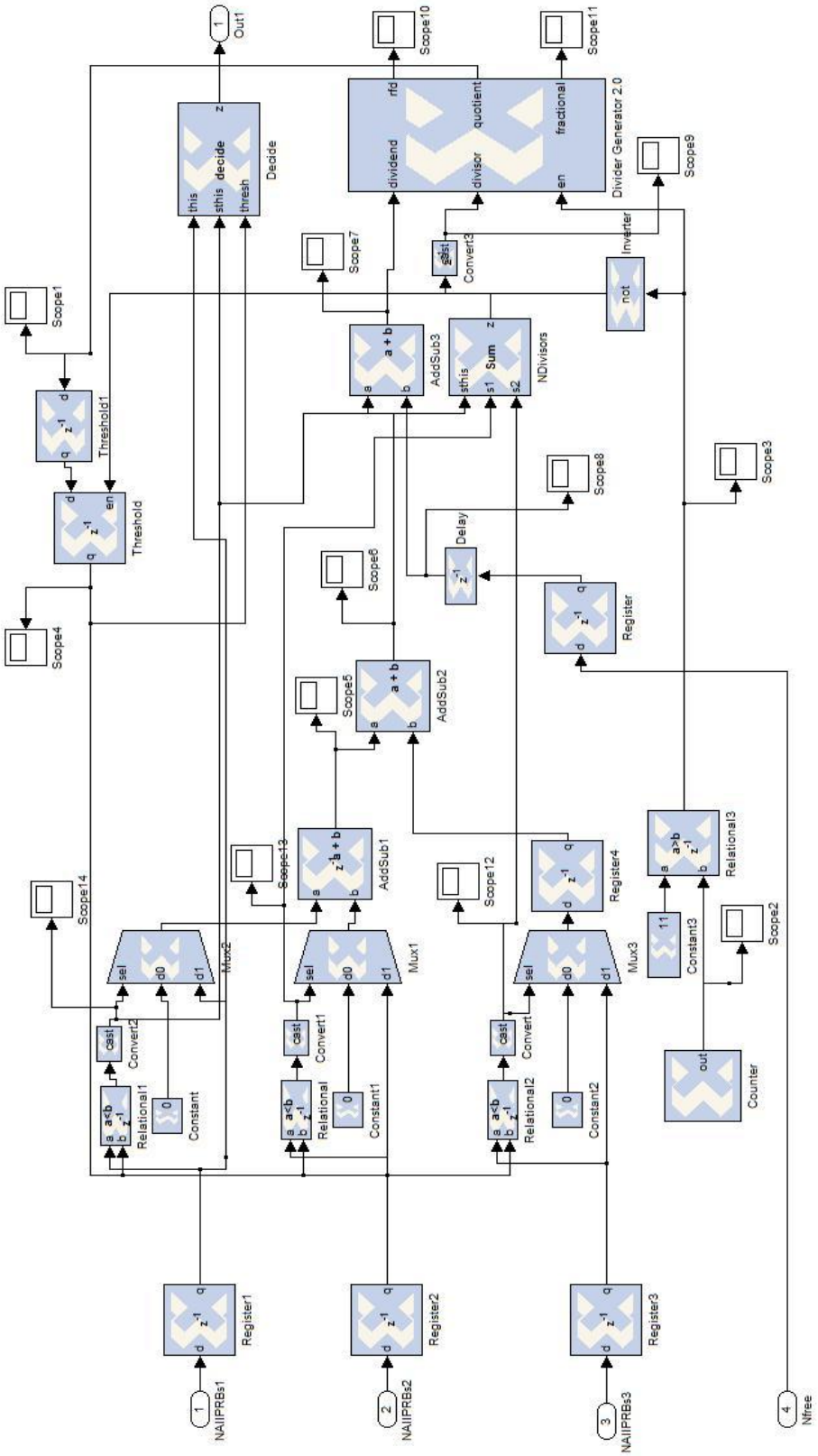

FIGURE 42 – THE FREE PRBS FAIR ALLOCATION BLOCK SCHEME

The implementation of this block is made under an important assumption: in this project the number of stations that interact on the channel is fixed at three. This limitation allows to simplify the computation needed to obtain the number of allocable subchannels and reduces the complexity of the system since the number of active devices is known.

The fair allocation computation is implemented following three considerations:

1. The performed operations are: sum of the valid inputs and then a division. The result is a threshold that has to be compared with the inputs in order to validate the inputs required for the next loops.
2. To perform the required operations, a single “sum system” composed by three AddSub blocks and a divider are implemented. Additional control blocks are then inserted to manage the computation loops. The implementation of a more complex control system, with a limited “computational” structure, is necessary in case of a greater number of active stations involved in the computation.
3. The final result of the computation will be available after a certain amount of time: a time control is needed to enable the output as soon as the value is valid.

In the computation loop a series of sums is performed before a division. The starting input values are the number of allocated subchannels of each active station and the number of free subchannels.
A series of multiplexers manage the inputs of the AddSub blocks: when a starting input is no longer valid (e.g. does not satisfy the threshold condition) the value sent to the AddSub is zero. This choice allows to sum together only the valid inputs at each iteration.
The final result of the sum is then divided by the number of total active station.

The number of active stations considered in the loop is the number of stations which satisfy the threshold condition. The output of the threshold comparison is then converted from boolean to unsigned and used to compute the number of active stations since an active station is represented by value “1” while an inactive station, by value “0”.

The new threshold value is defined by the output of the divider block (Figure 43): the division is a complex hardware operation that requires dedicated resources. In the implemented solution, the Divider Generator block performs a division using the Radix-2 algorithm setting: this options is recommended by Xilinx for operands width less than 16 bits [17].

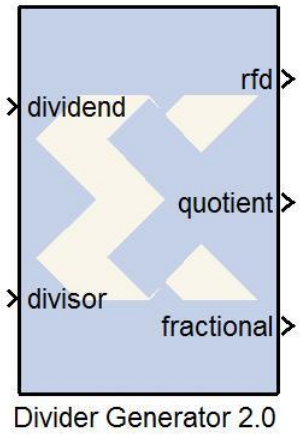

FIGURE 43 – DIVIDER GENERATOR 2.0 BLOCK

The final value of the fair allocation process, is the output of the Decide block (Figure 44). This block is generated from the provided MCode block of the Xilinx System Generator Library. The MCode blocks allows to incorporate a simple Matlab function into a Xilinx block to describe the device behavior at a higher abstraction level. In this case the implemented function allows to simplify the model scheme concentrating the desired functionalities in a single block. The MCode block performs the same operations performed by other System Generator blocks but reduces the design complexity.
The Decide block evaluates the threshold condition of the running station and sets the output to zero (no free subchannels can be allocated) if the local running station is above the fair allocation threshold.

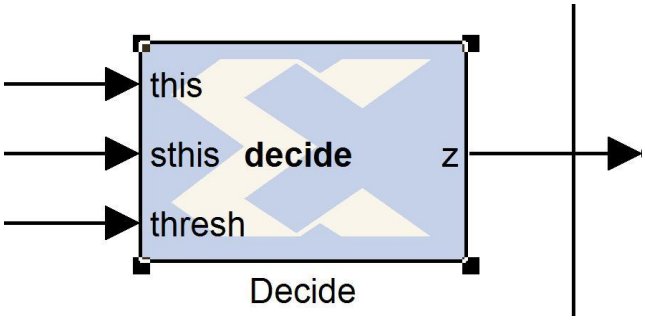

FIGURE 44 –THE IMPLEMENTED DECIDE BLOCK

3.3.3 ALLOCATION CONTROL

In order not to occupy more resources than what really needed by the station traffic requirements, the fair allocation output is limited by the Allocation Control block.

This control block performs two operations: limits the free subchannels allocation to the station needs and computes the number of additional subchannels needed if the free subchannels allocation does not fulfill the station requirements.
The Allocation Control block is shown in Figure 45.

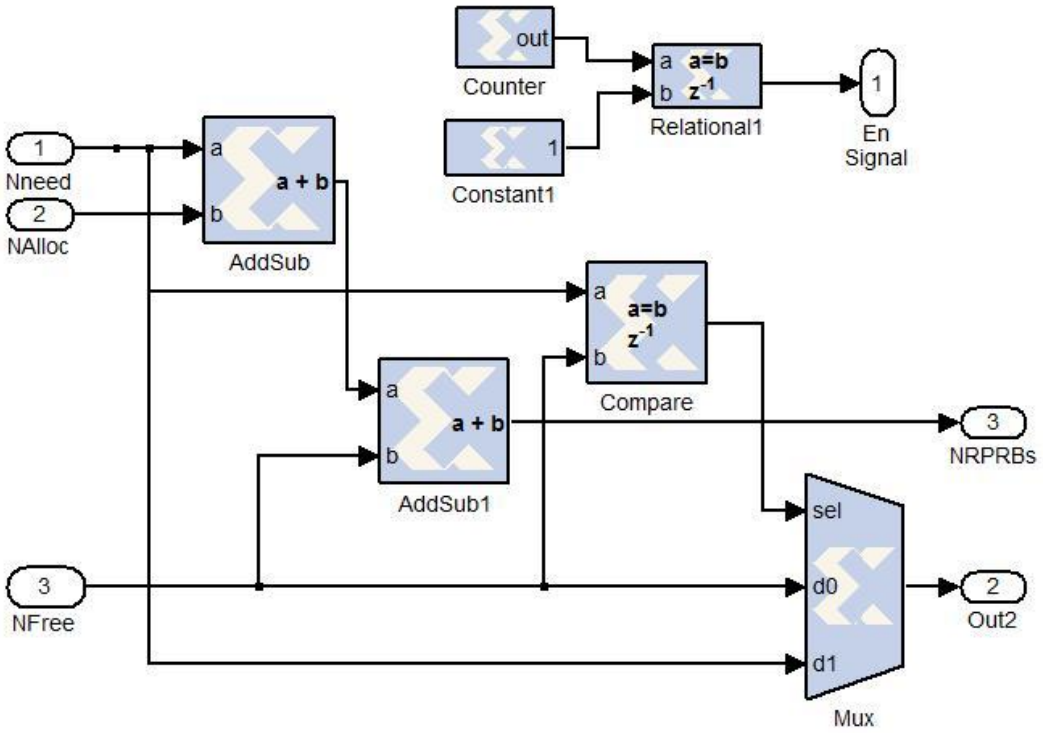

FIGURE 45 – ALLOCATION CONTROL BLOCK

AN ADDITIONAL COUNTER IS IMPLEMENTED TO MANAGE THE ENABLING OF THE ALLOCATION VECTOR CREATION WHEN A VALID FREE SUBCHANNEL VALUE IS AVAILABLE.

3.3.4 ALLOCATION VECTOR CREATION

After the computation of the required to-be-allocated free subchannels, the allocation vector is created.
The vector creation process follows the same principle of the high level implementation proposed in 2.5.2 : the interference power vector is sorted in order to choose the less interfered subchannels and then the previously defined number of free subchannels is allocated. A 48-elements binary vector is generated: an allocated subchannel is set to “1” while an unallocated subchannel is set to “0”. The allocation vector is then sent to the subchannel allocation block in the OFDM transmitter.

The high level implementation in Matlab/Simulink of such functionalities is quite easy: few lines of M-code allow to manage the 48-elements vector and set the appropriate values at the desired vector positions.

However, the System Generator level implementation is not so intuitive mainly due to the fact that the arrays cannot be handled directly by the system and that the provided mux/demux devices do not manage more than 32 inputs. The implemented solution scheme is depicted in Figure 46 and described in what follows.

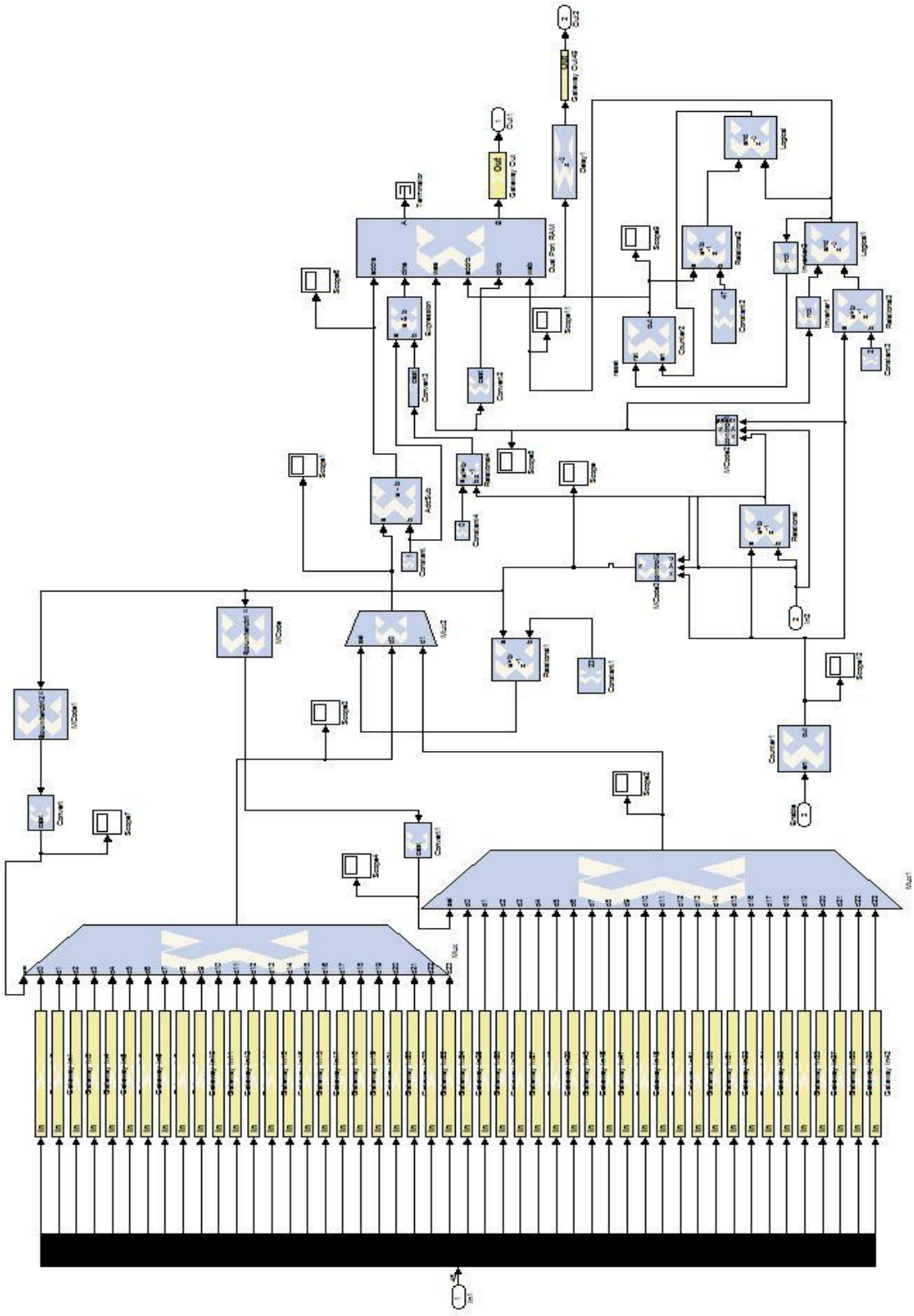

FIGURE 46 – ALLOCATION BLOCK

The Allocation Block manages the 48 elements of the interference power vector as inputs. It is assumed that the vector is sorted by placing the less interfered subchannels on the top of the array. What actually is given in input are the array positions of the less interfered subchannels and not the interference power values.
The 48 elements are then divided in two groups of 24 elements each, in order to be handled by the two multiplexer blocks.

The design concept is the implementation of the following steps:

1. Given the number of free subchannels N (previously calculated), the positions of the first N subchannels are read.
2. The read positions are used as addresses to access a Random Access Memory (RAM) block with a depth of 48 elements.
3. The RAM is initialized with zeros. Every time a RAM address, with the position of a to be allocated subchannel is given, a constant value “1” is written in the RAM.
4. As soon as the required number of free subchannels is allocated, a read-process of the whole RAM cells is performed: the values stored in the RAM are given in output as data stream. After the reading, the RAM cell is set to “0” in order to be ready for the next writing operation.

The implemented block requires some control functionalities to manage the vector scanning and the memory-access phases. The system operations can be divided in two phases :

1. Reading address values and access to the RAM memory cells.
2. RAM reading and allocation vector data stream creation.

Since the vectors cannot be directly handled, all the internal operations are converted to serial communications between the devices. A serial transmission of vector values requires a certain amount of time to be performed.
The whole resource allocation process has to be performed within the scheduled time interval so it becomes a main issue not to waste time and optimizing the involved processes.

A first counter is utilized to scan the inputs: when the required number of free subchannels is allocated, a second counter is enabled to perform the reading phase. The

management of counter operations has been a critical issue during the implementation the aim was to reduce the delays between the two operational phases.

The first counter (depicted in Figure 47) receives the enable signal from the Allocation Control block, as soon as the free subchannels value is valid.

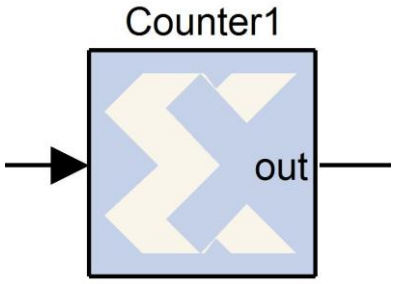

FIGURE 47 – SYSTEM GENERATOR PROVIDED COUNTER

As the counter increases its output value, the corresponding input port is selected and the vector position information is loaded as address for the RAM access.
A system composed of MCode blocks, comparators and Multiplexers is implemented in order to control the input ports selection. Only one input port value can be read as address in a single clock interval.

The key blockset component used in this block implementation is the Dual Port RAM device (Figure 40).

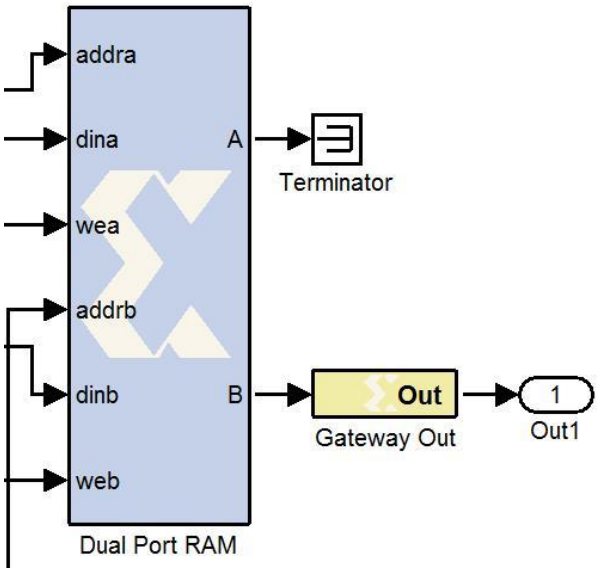

FIGURE 48 – IMPLEMENTED RAM BLOCK
OUTPUT A IS NOT USED AND CONNECTED TO A TERMINATOR.

The Dual Port RAM block implements a RAM where dual ports enable a simultaneous access to the memory space [17]. The dual access is preferred in the system design since it requires less access control logic.
The block has two independent input sets: every set is composed by a address port (addra,addrb) used to access a specific memory area, a data port to provide data (dina,dinb) and an enable port that enables the inputs (wea,web). In this project implementation, the input set A is used for the writing operations while the input set B is used for reading procedure. The A and B ports are active when the enable signal has high logic value.
Output port A is not used since no data is read from the memory while the "allocation" phase is performed. Port B instead, outputs the final vector data stream.
The write enable A port is indirectly driven by the first counter that manages the writing process while the write enable B port is driven by the second counter that manages the reading process. The main difference between the first and the second counter is that the first counter is disabled when the desired number of subchannels is allocated, while the second counter always performs a 48 steps operation to read the whole memory area.
The memory area is addressed with values that vary from 0 to 47: the position values in the interference power vector instead, vary from 1 to 48 so an address control has to be implemented.
The dina port is used to write the value "1" in each allocated vector position: if no subchannels are required to be allocated, the value is set to "0" by default.

3.4 COMPLETE SYSTEM

The implemented algorithm functionalities receive in input the sorted interference power vector positions together with the values of subchannels allocation provided by the other stations. The output is a serial data stream that contains the values of the binary allocation vector. In order to be utilized in the Simulink simulation model this serial data stream is then buffered and sampled. In this implementation the buffering and sampling operations are implemented with Simulink blocks.
The simulation results performed in order to validate the functioning of the developed system are presented in Chapter 4, Section 4.3.

4. RESULTS

4.1 VALIDATION

In the previous chapters a description of the design process and the implemented solution is provided. The following sections of the report focus on the validation of the system and analyze its behavior in both Simulink and System Generator implementations.

4.2 VALIDATION OF THE IMPLEMENTED MATLAB/SIMULINK MODEL

In Chapter 2 a description of the design solutions adopted for the system simulation, together with a description of the transmission environment (2.3) are provided. The simulations performed in Matlab/Simulink are intended to provide validation for the implemented system behavior in different resource allocation setups.

The scope of the simulations is to analyze how the implemented algorithm performs in terms of average throughput, average SNR at the receiver, and number of allocated subchannels. In order to validate the implemented solutions, the simulations are expected to provide:

- a balanced allocation of the subchannels according to a pre-defined resource requests.
- a reasonable SNR value in the range of the values indicated in Chapter 2, Section 3.

The average throughput is computed on a frame basis: the received SNR defines the used modulation scheme and the throughput value is obtained multiplying the number of bits transmitted by the selected modulation with the number of utilized subchannels.
The pre-defined amount of subchannels request for each station, is the main parameter that "stresses" the algorithm performance: increasing the request amount of subchannels, pushes the allocation functionalities to constantly require more resources and increases the competitiveness between the active stations. Modifying the initial subchannels request it is possible to evaluate how the algorithm reacts and how it manages a balanced transmission avoiding interference.

In this report, the results of three significant simulations are presented:

- a simulation in a "balanced" context where the three systems have an equal and balanced subchannels request of 16 subchannels each
- a simulation in a "competitive" context where the three systems need to allocate 25 subchannels each and the algorithm needs to balance the greedy request
- a simulation in an "unbalanced" context where the stations have different resource requests.

Additional simulations have been performed to validate the system in less competitive context but the results are not significant for this analysis.

The results are obtained after performing an average on the values of 100 simulations for each of the various context settings. The number of simulations has been estimated evaluating the model behavior in different initial random allocations and defined in order to have statistically significant values. The simulation time is chosen after an evaluation of the required time by the algorithm to stabilize the allocation values. According to this evaluation, a simulation time of 0.2 sec is sufficient to have reasonably stable subchannels allocation values. The subchannels allocation values are obtained averaging the allocation values obtained after the last resource allocation phase performed by the system in the simulation time.
The throughput of the system is calculated on a burst basis, considering two OFDM symbols transmitted per burst. The systems works in "full buffer" mode for the transmitted data. The maximum throughput in a full band case with a 64-QAM ¾ modulation, is 1728 bits per burst.

4.2.1 "BALANCED" CONTEXT SIMULATION

The purpose of this simulation is to show the system behavior in a relatively competitive context. In this case every station needs to allocate 16 subchannels over the total amount of 48. It is expected that system achieves a fair allocation of 16 subchannels per each station.

	System 1	System 2	System 3
Subchannels request	16	16	16
Allocated Subch. average over 100 sims	15,4242	15,3939	15,4949
Average SNR at receiver (dB)	14,6367	16,6745	18,3544
Average Throughput (bit/burst)	227,0558	235,8458	276,0128

TABLE 5 – SYSTEM PERFORMANCE RESULTS IN A "BALANCED" CONTEXT

Table 3 shows the results obtained after running 100 simulations.
The allocated subchannels average values are around 15.4 in all three cases. This first result leads to two considerations: the algorithm allocates fairly among the three stations showing an equal amount of subchannels for each, the algorithm settings are cautious and do not allocate the 100% of the available subchannels.
The SNR values at the receiver are reasonable: the free transmission SNR values obtained in Chapter 2 Section 3 were 15dB,16.5 dB and 19 dB respectively.
In this "balanced" simulation, the three stations have equal values for the subchannels allocation but System 3 has an evident better average SNR value. The threshold for enabling 16-QAM ¾ is 19dB: System 3 can switch modulation and achieve a better average throughput performance.

4.2.2 "COMPETITIVE" CONTEXT SIMULATION

The second simulation presented in this report aims to analyze the system performance in a "competitive" context: the three stations have an equal resource allocation request, of 25 subchannels. Given the total number of 48 subchannels available on the channel, it is evident how such resource request leads to a competitive environment where each station tries to allocate a number of subchannels greater than the fair value (16 subchannels). Since the three active station have an equal resource request, the algorithm is expected to allocate an equal amount of 16 subchannels.
Table 4 provides the simulation results for the "competitive" context.

	System 1	**System 2**	**System 3**
Subchannels request	25	25	25
Allocated Subch. average over 100 sims	16	17,4343	15,8383
Average SNR at receiver (dB)	14,5176	16,0442	17,9662
Average Throughput (bit/burst)	251,2439	296,4489	272,3388

TABLE 6 – SYSTEM PERFORMANCE RESULTS IN A "COMPETITIVE" CONTEXT

The allocated subchannels average value gives an important information about the behavior of the algorithm: given the greedy resource request by each station, the system manages to allocate almost the expected amount of subchannels. The average allocation for System 1 is exactly 16 subchannels, for System 3 is 15.8 while for System 2 is over the target value at 17.4. In this "competitive" context, the system is pushed to continuously allocate subchannels. After the first couple of allocation rounds it is reasonable to assume that not enough free subchannels are available for allocation: the pre-defined Low Interference threshold becomes a decisive parameter in the allocation of LI subchannels. In this simulation a value of -7dB of interference power on a single subchannel is chosen.
The SNR average value for System 1 is equivalent to the previous simulation.

The SNR of System 2 and System 3 is circa 0.5dB lower to the previous case: given the small interference generate by the overallocation of System 2 it is reasonable to relate the 0.5dB loss to the mutual interference between System 2 and System 3.
The throughput results are consistent with the SNR and subchannel allocation values.

4.2.3 "UNBALANCED" CONTEXT SIMULATION

Another interesting case for system validation is to test the algorithm with an "unbalanced" resource request. Table 5 shows the results obtained with an allocation scheme of 25x16x10. The total amount of requested subchannels, exceeds the available number by 3.

	System 1	**System 2**	**System 3**
Subchannels request	25	16	10
Allocated Subch. average over 100 sims	20,6364	14,5353	9,75756
Average SNR at receiver (dB)	14,6367	16,6745	18,3544
Average Throughput (bit/burst)	324,4350	223,5426	169,4318

TABLE 7 – SYSTEM PERFORMANCE RESULTS IN A "UNBALANCED" CONTEXT

The average allocated subchannel values shows that the systems tends to underallocate the subchannels. Since the total subchannels request exceeds the available number, it is reasonable to expect all the station to reduce their amount of subchannels; the results provided by the simulation show that the most of the required reduction is performed by System 1: System 2 gets an average value of 1.5 subchannels less than the target value while System 3 almost gets all the required subchannels.
The achieved SNR values are in the expected range showing that if the algorithm manages to avoid the mutual interference, the achieved SNR at the receiver is acceptable.

The provided results shows that the implemented algorithm manages to allocate the subchannels avoiding the interference in the “balanced” and “unbalanced” cases, the allocation process generates a limited interference in the “competitive” simulation.
The SNR results are good and consistent with the free transmission obtained values.
A more detailed result analysis is given in 4.5.

4.3 Validation of System Generator implementation

After the analysis of the implemented system design in the Matlab/Simulink environment, a validation of the System Generator implementation is performed.
In this project, the implemented solution has been tested in a stand-alone setup where appropriate input values are provided. This section of the report describes the system behavior providing the scope graphs of the signals in various points of the implemented setup.

4.3.1 Test A

In the first test a basic allocation situation is evaluated: it is assumed that the running station has allocated 10 subchannels under a request of 16. It is also assumed that the other two stations have allocated 16 subchannels each. Given the fair allocation algorithm, it is expected that the Free PRBs block allocates the 6 best subchannels in the allocation vector.

Since the position of the subchannels is not relevant for the system analysis, the interference power vector provided is manually edited in order to have significant values in output. In this simulation the interference power values are modified in order to have an allocation of the first and the last subchannels position. This choice is made in order to have a better understanding of the output vector stream.

As previously described, the first operation is the computation of the number of free subchannels on the channel. In this case it is provided a specific input vector that has 6 subchannels under the defined threshold. Figure 49 shows the output of the NFreePRBs block: after 1 clock period delay given by the threshold comparisons, the output is set to 6 and given in input to the next block.

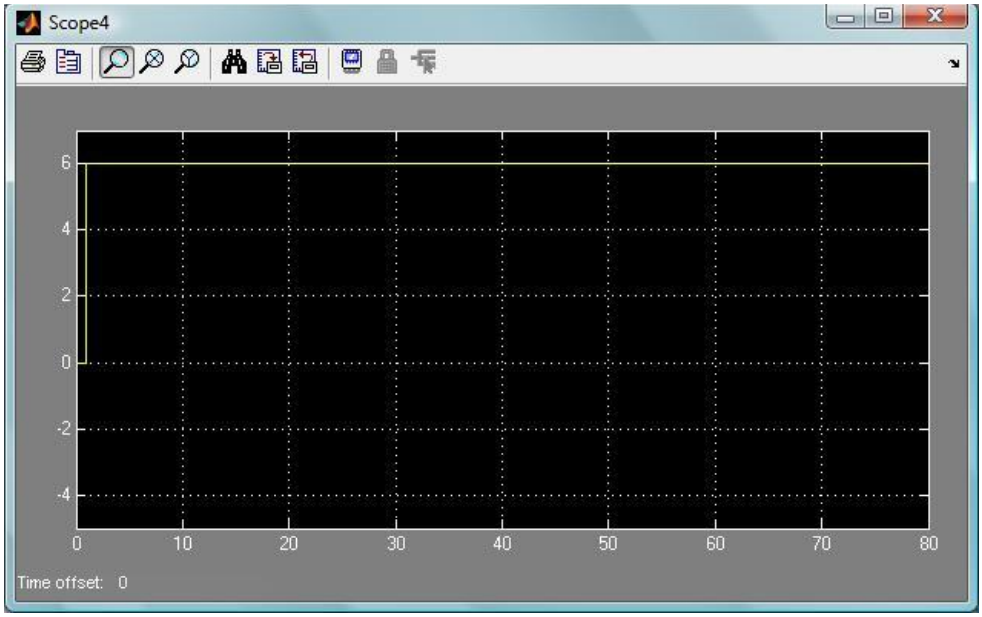

FIGURE 49 – NFREEPRBS OUTPUT SIGNAL

The number of free subchannels is needed to perform the fair allocation computation. This block requires a certain amount of time before the output value can be considered valid.

Figure 50 shows how the expected number of allocated subchannels for the running station, is 6: the computation delivers the value after 13 clock periods since the system needs to compute only one threshold.

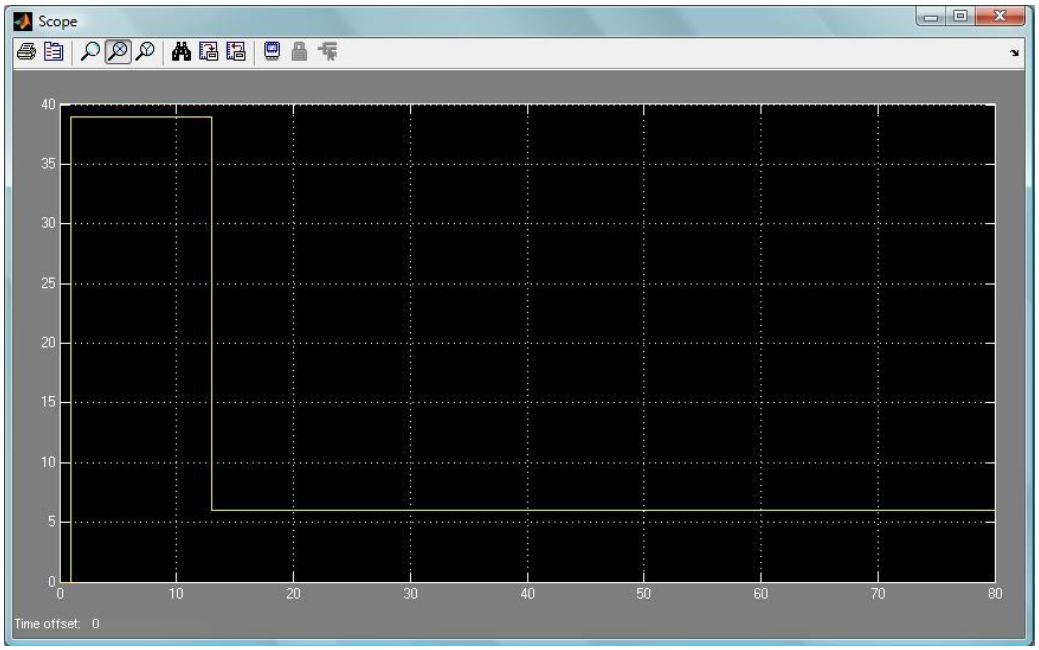

FIGURE 50 – FREE PRBS FAIR ALLOCATION OUTPUT (1)

4.3.2 TEST B

In this second test, a more complex allocation context is defined in order to highlight how different computation requirements produce significant variations in terms of delay.
The running station request is always of 16 subchannels with a previous allocation of 10. In this test, System3 allocates 18 subchannels, System2 is constant to 16. The total amount of free subchannels is 4. We expect the system to allocate 4 subchannels.
The chosen positions for the subchannels allocation are at number 1,3,24 and 48 in order to visualize the beginning and the end of the vector data stream.

Figure 51 shows how this new setup causes a longer computation in the Free PRBs Fair Allocation block. In test A, System2 and System3 were immediately excluded by the fair allocation since they already allocated a number of subchannels above the fair allocation threshold. In test B instead, only System3 is immediately excluded and a new threshold is computed in an additional computation loop. Figure 51 shows that the desired fair allocation value is valid after 37 clock periods.

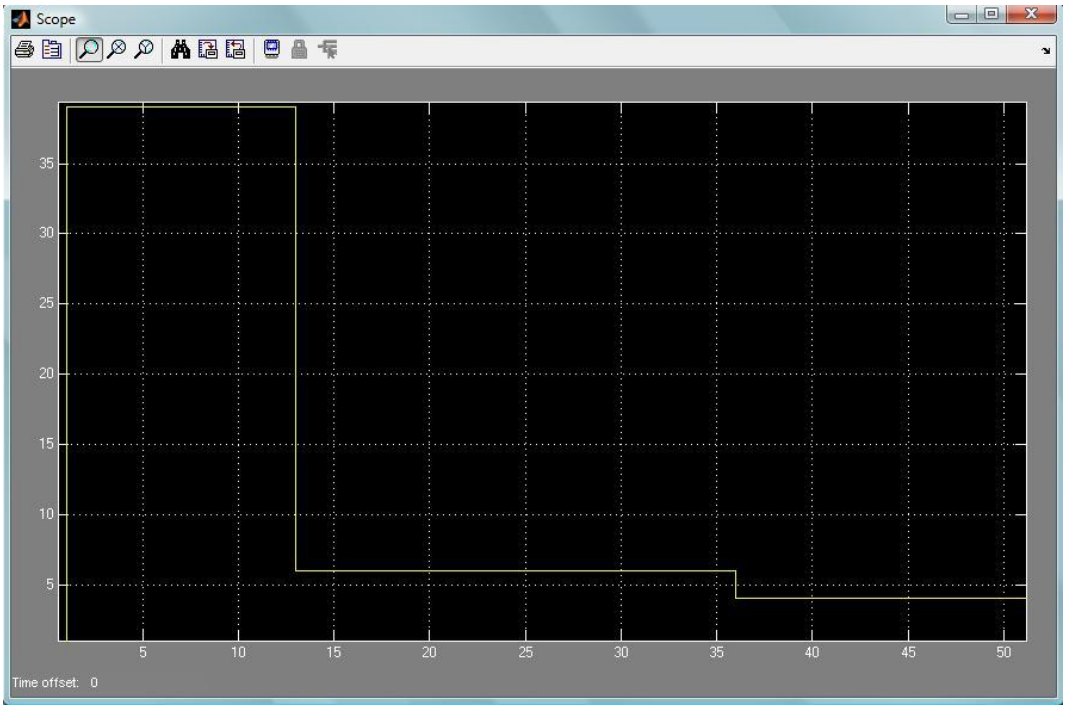

FIGURE 51 –FREE PRBS FAIR ALLOCATION OUTPUT (2)

The assumption of having a fixed number of active systems limits the time needed to have a valid fair allocation value. This system implementation requires more computation time when the number of stations to be managed, increases.
As previously described, the Allocation Control block, manages the enabling of the vector creation when the number of free subchannels to be allocated is valid. When the Allocation block is enabled, the allocation vector process begins.
Figure 52 shows the output of the Allocation block with the Test B settings.

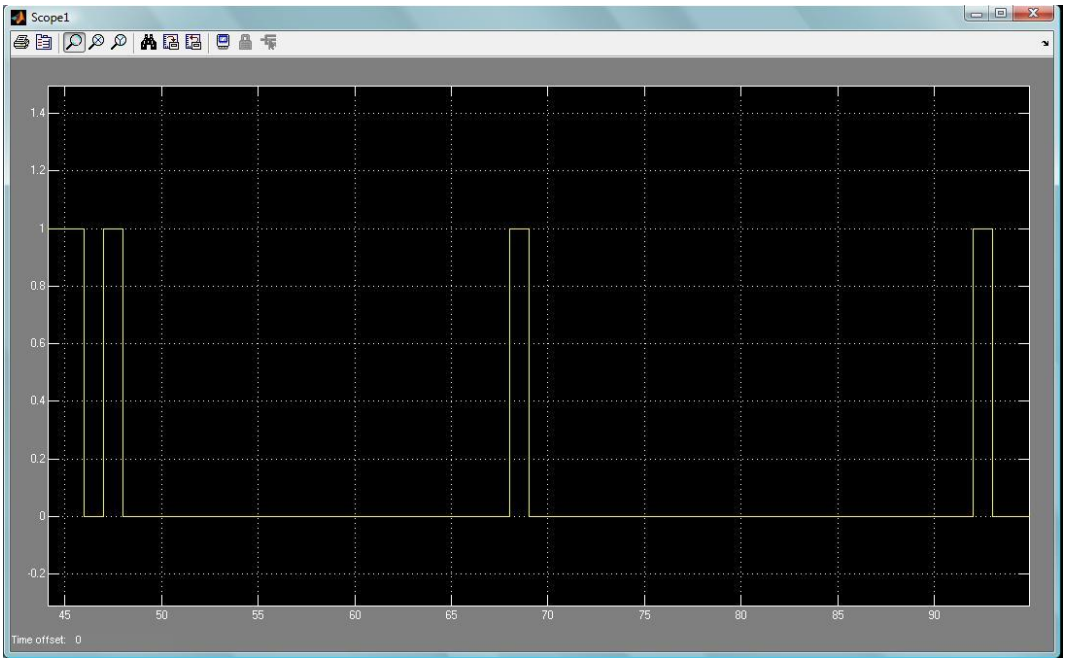

FIGURE 52 – ALLOCATION BLOCK OUTPUT

THE SIGNAL REPRESENTS THE BINARY VALUES OF THE ALLOCATION VECTOR.

The signal depicted in the scope in Figure 52, represents the stream of logical values composing the allocation vector. The first value is valid after 45 clock periods since the beginning of the simulation time, the last value is given 48 periods later as expected. As the signal scope shows, 4 subchannels have been allocated in the previously defined positions.

4.3.3 TEST C

In this test the number of allocated value of the running system is set higher than the fair allocation threshold: as expected, after one computation loop, the number of free subchannels to be allocated is zero and the output allocation vector is a stream of zero values.

4.3.4 TEST D

Another test is made in order to understand the system performance in a “worst case” situation.
The time required for allocating the subchannels writing the values in the RAM memory increases with the number of subchannels to be allocated. The worst case is then when there are 48 free subchannels and the running station is allowed to allocate all of them.
Figure 53 shows the allocation vector data stream given in output: the last value of the vector (the vector is composed only by “1” elements since every subchannel is allocated) is processed after 137 clock periods.

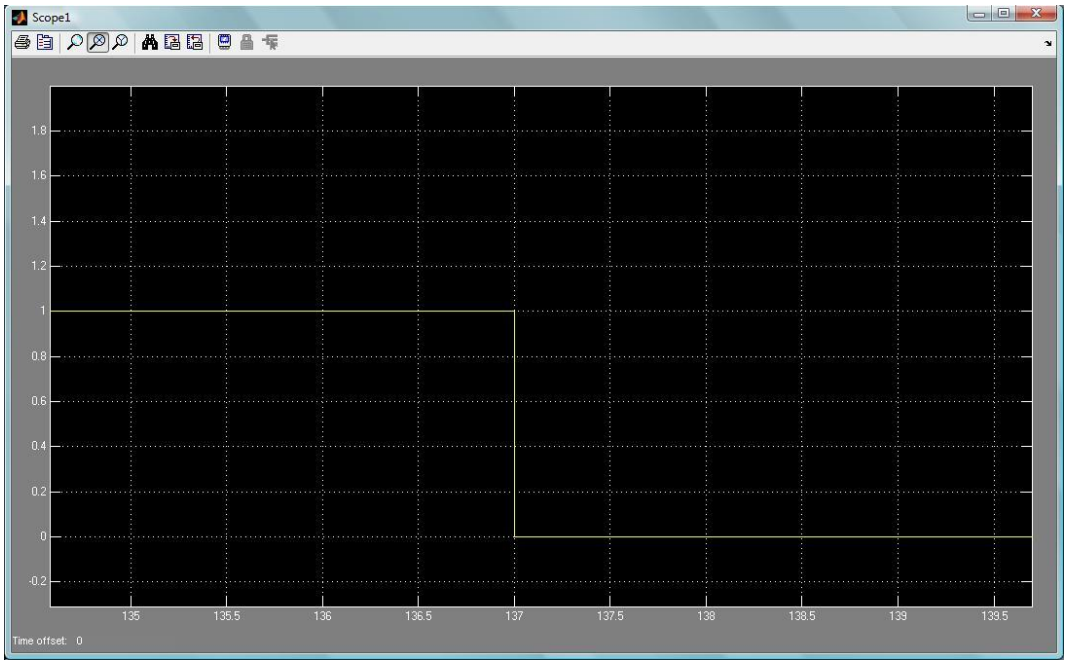

FIGURE 53 – DETAIL OF THE OUTPUT DATA STREAM IN THE WORST CASE SITUATION

The provided system analysis validates the behavior of the implemented blocks that work correctly.

4.4 HARDWARE RESOURCES ANALYSIS

After the implementation and the validation of the system an analysis in terms of hardware resources on FPGA can be performed.
Xilinx System Generator provides two tools for resource computation: the Resource Estimator block and a more detailed analysis in the compilation report. The detailed

report is obtained first compiling the implemented blocks with System Generator and then performing a synthesis with Xilinx ISE software [17].

4.4.1 Resource Estimator

The Resource Estimator block provides a fast method to estimate the FPGA resources required for implementing a System Generator model or subsystem.
The estimates are performed by invoking block-specific estimators for Xilinx blocks, and summing these values to obtain a global estimate [17].

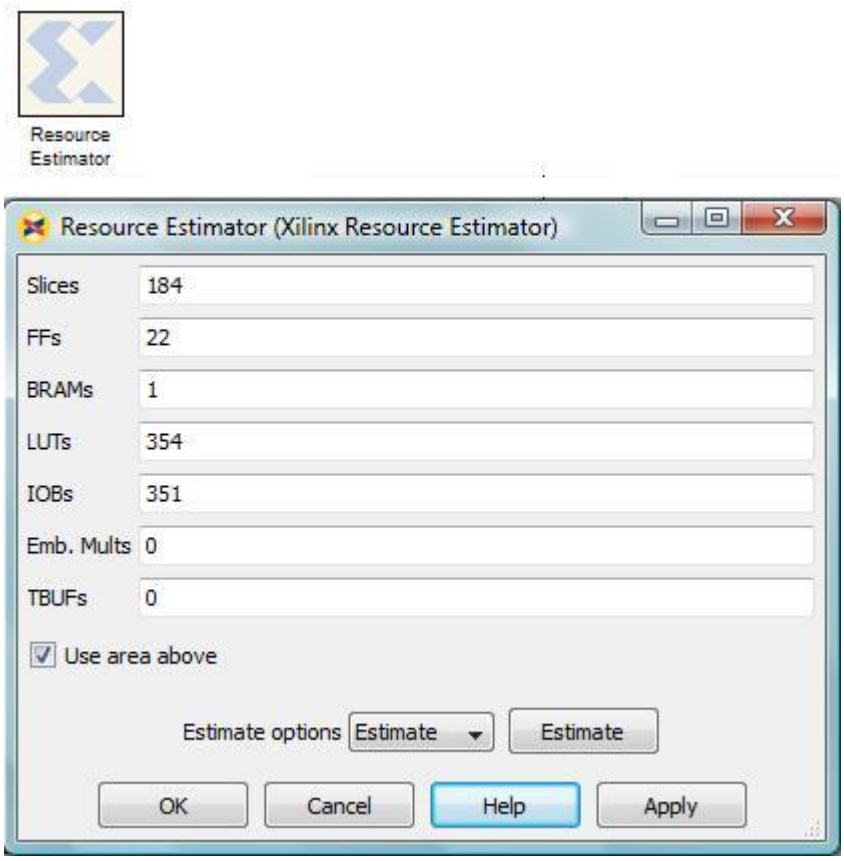

Figure 54 – Resource Estimator block values

Figure 54 shows the estimates results for the implemented Free PRBs Allocation block.
The Resource Estimator provides estimates in terms of Slices, FFs (Flip Flops), BRAMs (RAM blocks), LUTs (Look-up tables), IOBs (Input/Output blocks), Embedded Multipliers and Tristate Buffers (TBUFs).
The resource estimation performed with the “Estimate” setting, provides only rough numbers: a more accurate analysis is therefore performed with the System Generator HDL compilation and ISE synthesis process.

4.4.2 System Generator Compilation results

The ISE synthesis tool allows to generate an accurate report synthesizing the HDL code. The system compilation is performed considering the target platform: in this project a Virtex5 xc5vsx50t-1ff1136 is considered.

xilinxblockunified_cw Partition Summary	[-]
No partition information was found.	

Device Utilization Summary (estimated values)				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	
Number of Slice Registers	247	32640		0%
Number of Slice LUTs	649	32640		1%
Number of fully used LUT-FF pairs	0	896		0%
Number of bonded IOBs	615	480	128%	
Number of Block RAM/FIFO	1	132		0%
Number of BUFG/BUFGCTRLs	1	32		3%

Figure 55 – Device utilization values

This table reports the estimated values obtained with the System Generator compilation

From the values reported in Figure 55 it is possible to visualize that the implemented system does not occupy considerable resources on the FPGA (between 0% and 3%) except for the Input/Output blocks (IOBs). In this case a usage of 128% of the available resources, is reported.
The report results indicates that the rough estimation provided by the Resource Estimator block in “estimate” mode, significantly underestimates the required resources with almost halved values

The value of the I/O blocks is mainly given in its greatest part, by the number of input/output gateways multiplied by the bit precision of the signal. A group of 48 signals with a required precision of 6 bits each, occupies 288 blocks.
The high number of IOBs reported is due to the fact that the standalone setup implemented in the project, uses a double input vector for managing the simulations. This is a simulation oriented choice that has not a real impact on the implementation of the system on the FPGA. A different setup is however needed to download the firmware on the board.

The fully implemented system, completed with the sorting block, needs a single power interference vector of 48 elements with a precision of at least 7 bits for a total amount of 336 IOBs. Even if an optimal system design requires almost half IOBs, the number of input ports represents by far the most critic parameter in the system implementation.
Future work is expected to provide an effective solution to the input ports management by modifying the simulation setup and developing a sorting block that handles the required signals by using a single vector input.

Another important parameter that is provided in the compilation report is the operative frequency of the system. The results indicates:

- Minimum clock period: 12.362ηs
- Maximum frequency of 80.893 MHz

The defined period for the resource allocation phase is set to $3.6 * 10^{-3}$s.
Considering the previously defined "worst case" situation, where the implemented blocks require 137 clock periods to deliver the allocation vector stream, the total time with the estimated minimum clock period is:

12.362ηs*137 = 1.69μs. (4.4.2.1)

4.5 Results Evaluation

The reported model simulations and tests allow to analyze the system performance, the resource requirements for hardware implementation and to individuate the main issues related to the optimization and possible improvements to the implemented system.

The analysis of the Matlab/Simulink simulations on the implemented system model, gives the following indications:

- The implemented algorithm operates correctly and performs the resource allocation among the three involved systems avoiding the mutual interference: the achieved SNR values at the receiver are satisfy the expectations.
- Although the algorithm manages to perform a fair subchannels allocation, an optimization process is needed. The system does not achieve a 100% usage of the available subchannels in the "balanced" context.
- In the "competitive" context the system tends to overallocate over the fair allocation value and generates a small amount of interference. The overallocation might be managed modifying the LI threshold.

- In the “unbalanced” context the system is too cautious in the allocation, the result is a waste of resources. Furthermore, the results show that the reduction of exceeding subchannels request is mainly distributed on the system with the highest resource requirements. This system behavior is mainly due the fact that the algorithm tends toward an equal and fair allocation among the active stations: in free subchannels allocation, the systems privileges the stations with subchannels requirements below the fair allocation threshold.

As regards the System Generator implementation, the analysis of the scope signals provides a validation of the system functionalities.
In the hardware implementation the main design concern is about computation time and signals synchronization among the involved blocks. In terms of required computation time the assumption of having three active stations, simplifies the fair allocation process and ensures a defined computation delay.
The hardware resources required to perform the computation processes are not high demanding, only basic logic operations are performed: in the fair allocation block, the divider is the most complex component and generates the most significant delay.
The implemented system could be intended as a chain of subsequent processes that require a certain amount of control logic in order to be properly managed. In order to translate the algorithm behavior at hardware level and implement the various computational steps, the design of the signal control blocks is the key issue.

The results provided by the resource estimations, indicate that managing an input vector of 48 signals has a great impact in terms of Input/Output resources on a FPGA platform. The implemented system requires the implementation of an additional block that manages the interference power vector, in order to be executed on the FPGA. 48 signals with 6 bits precision indeed, require 288 IOBs: avoiding the usage of the additional vector, the total IOBs request will suit the FPGA available resources amount.

The estimated operational clock period, allows the implemented block to operate in a relatively short time compared to the system interval assigned for the resource allocation phase. The complete Decision Making in order to operate correctly, implements a vector sorting block that is expected to affect considerably the required hardware resources and the total computation time. A complete hardware analysis has to be performed considering the implementation of the sorting block.

5. CONCLUSIONS

5.1 PROJECT DEVELOPMENT SUMMARY

The purpose of this project was to analyze the implementation issues of a dynamic spectrum allocation algorithm.

Next generation wireless communication networks are expected to achieve high data rates using a wide spectrum allocation in the range of 100MHz: given such wide allocation it is also expected that various operators will coexist on a shared channel in a Local Area. This assumption is at the base of the research on the Flexible Spectrum Usage that addresses the resource allocation problem in the previously described context by the use of the Spectrum Load Balancing algorithm.

The analysis of the implementation issues of a modified version of SLB is the scope of this project.

In order to implement a version of the allocation algorithm, a suitable simulation model has been individuated. Since the proposed algorithm works on OFDM transmission systems, the first task was setting an OFDM-based transmission environment model that allows to investigate the algorithm implementation issues. In the project development there were no restrictions about the communication standard to use, given the fact that the algorithm core functionalities can be tested on many OFDM-based systems. The IEEE® 802.16-2004 OFDM PHY Link, Including Space-Time Block Coding model provided in the Mathworks Matlab/Simulink simulation environment, showed suitable model features and has been chosen as the starting basic structure for the project development.

Following the indications of the proposed Rugby methodology, the implementation analysis has been structured on different level of abstractions: at first a Matlab/Simulink implementation has been performed and after the validation of the system, a low level implementation has been realized with the utilization of Xilinx System Generator.

The realization of the Matlab/Simulink model has been performed by first modifying the starting WiMAX scheme, in order to simulate the transmission of three stations to separate receivers, and in a second time, implementing the specific algorithm functionalities in dedicated model blocks.

The validation of the implemented Simulink model has been performed by running several simulations with different allocation settings. In order to analyze the system behavior, the simulation tested the performance of the implemented algorithm block in a “balanced”, “unbalanced” and “competitive” allocation setups.

The provided results validate the system implementation showing that the systems avoids the mutual interference of the transmitting station allowing the usage of a shared channel.

The SNR and throughput results in all the simulations demonstrate that the system achieves reasonable values (15-19 dB) in the range of the values obtained running a free transmission by a single station.

The subchannels allocation results, show that the systems does not achieve a 100% usage of the subchannels in the “balanced” context and in the “unbalanced” context, in the “competitive” context a small amount of interference is generated due to an overallocation by one subchannel.

After the validation of the implemented Matlab/Simulink model, the core functionalities of the algorithm specific blocks have been re-designed with Xilinx System Generator aiming to implement the system on the target FPGA platform.

Due to the short time available, the System Generator design process did not consider the implementation of all the individuated blocks and focused on the Free PRBs Allocation block that implements the main elements of the resource allocation algorithm.

The implemented system has been then analyzed in order to understand the delay generated by the performed computations. The system has been realized in a stand-alone configuration in order to perform the necessary validations.

The compilation results generated by the System Generator and by the Xilinx ISE synthesis give detailed information about the resource requirements of the implemented elements. The implemented blocks do not have high hardware requirements on the target FPGA platform (0-3% usage) except for the Input/Output blocks (128%). The stand-alone setup generates an input ports requirement that exceeds the FPGA capacity.

5.2 SHORT-TERM PERSPECTIVES

The implemented System Generator features need to be tested in a complete environment. Since the available resources in terms of IOBs are limited, an alternative implementation solution is required in order to implement the system on the FPGA platform. There are two solution that can be implemented in short term period: the first option is to serialize the inputs, the second one is to implement a sorting block that manages the required inputs.

The input serialization, relatively easy to implement, significantly decreases the IOBs usage but increases the computation time of the system, adding a delay by reading the input serial values. The second option is to implement a sorting algorithm that receives in input the interference power vector, and generates the indexes vector and the interference value vector in output. The implementation of the sorting algorithm is a complex design process that requires a deep analysis and investigation. However the sorting block realization enables a complete evaluation of the hardware requirements and the performance of the Decision Making block that implements the algorithm functionalities.

5.3 LONG-TERM PERSPECTIVES

In long term period, an optimization process of the algorithm functionalities is desirable. An analysis of the implemented interference thresholds can lead to a better subchannels allocation and a lower interference in “competitive” environments.

A complete data management in relation to the allocated subchannels allows to compute the performed errors and better analyze the simulation model performance.

Other functionalities can be investigated, widening the analysis scope: for example considering a multi-user allocation on a single station enabling OFDMA.

The hardware implementation considered in this project only covers the core blocks of the resource allocation functionalities: future work may allow the implementation of other blocks, like the channel sensing and interference power computation, or the IFFT Input packing block in the OFDM transmitter.

REFERENCES

[1] Kumar, S., G.Costa, Kant, S., Frederiksen, F., N.Marchetti, & Mogensen, P. (February 2008). Spectrum Sharing for Next Generation Wireless Communication Networks. *Cognitive Radio and Advanced Spectrum Management. CogART 2008. First International Workshop on 2008.*

[2] Kumar, S., Wang, Y., Marchetti, N., Kovacs, I., Pedersen, K., & Mogensen, P. (2008). Spectrum Load Balancing for Flexible Spectrum Usage in Local Area Deployment Scenario. *Proceedings of the 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks. DySPAN 2008* .

[3] Cooklev, Todor. *«Wireless communication standards: a study of IEEE 802.11, 802.15, and 802.16.»* 1-4. IEEE Standards Association, 2004.

[4] Ohrtman, Frank. *«WiMAX Handbook.»* Blacklick, OH, USA: McGraw-Hill Professional Publishing, 2005.

[5] Jeffrey G. Andrews, Arunabha Ghosh, Rias Muhamed. *«Fundamentals of WiMAX: Understanding Broadband Wireless Networking.»* Chapter 2. Prentice Hall, February 2007.

[6] C. E. Shannon . Communication in the presence of noise*. Proc. Institute of Radio Engineers vol. 37 (1): 10–21.* January 1949.

[7] Moore, Gordon E. "Cramming more components onto integrated circuits" *Electronics Magazine. 4*. 1965

[8] Carl Shapiro and Hal R. Varian. "*Information Rules" 184. Harvard Business Press*. 1999.

[9] IEEE Standard 802.16-2004, *"Part 16: Air interface for fixed broadband wireless access systems"* ieee802.org October 2004.

[10] IEEE® 802.16-2004 OFDM PHY Link, Including Space-Time Block Coding, Copyright 2007 The MathWorks, Inc. Published with MATLAB® 7.6

[11] Xilinx Virtex-5 Family Overview. DS100 (v5.0) February 6, 2009.

[12] ML505/ML506/ML507 Evaluation Platform. User Guide. UG347 (v3.1) November 10, 2008.

[13] A. Gerstlauer, R. Dömer, J.Peng, D. Gajski *"System Design: A Practical Guide with SpecC"*, Kluwer Academic Publishers, 2001.

[14] Axel Jantsch, Shashi Kumar, Ahmed Hemani, *The Rugby Meta-Model*. Royal Institute of Technology, Sweden, 2000.

[15] Gajski, D.D.; Kuhn, R.H. "*New VLSI Tools"*, Computer Volume 16, Issue 12, Dec. 1983 Page(s):11 – 14

[16] H.Schulze,C.Lüders, "*Theory and applications of OFDM and CDMA*" ,John Wiley & Sons, 2005.

[17] System Generator for DSP, Release 10.1. Xilinx, March 2008.