UNIVERSITÉ DE MONTRÉAL

IMPLEMENTATION OF NEW MULTIPLE ACCESS TECHNIQUE ENCODER FOR 5G WIRELESS TELECOMMUNICATION NETWORKS

ZAHRA RAHMANI

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAITRISE ÈS SCIENCES APPLIQUÉE

(GÉNIE ÉLECTRIQUE)

AOÛT 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

IMPLEMENTATION OF NEW MULTIPLE ACCESS TECHNIQUES ENCODER FOR 5G WIRELESS TELECOMMUNICATION NETWORKS

présenté par : RAHMANI Zahra

en vue de l'obtention du diplôme de : Maitrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. DAVID Jean-Pierre, Ph. D, président

M. SAVARIA Yvon, Ph. D, membre et directeur de recherche

M. BOYER François-Raymond, Ph. D, membre

DEDICATION

To my family and beloved ones…

For making my earlier heaven more wonderful each day by being who they are

For their endless love, support and encouragement

ACKNOWLEDGEMENTS

One of the joys of completion is to look over the journey past and remember all the people who have helped and supported me along this long but fulfilling road. The present work would not have been possible without the valuable help of my academic advisor.

I would like to express my sincere gratitude to my supervisor, Prof. Yvon Savaria, for his support and valuable comments during my studies at Polytechnique Montréal. His guidance and patience smoothed the path and made it possible for me to develop this thesis. It was both an honor and a privilege to work with him. His feedback on this work brought new and interesting perspectives to the problem.

I must thank Dr. Normand Belanger for his professional advice and feedback during my research work. Our productive discussions helped me to elevate the quality of my research work. His passion for research, his patience, and his innovative ideas have impressed me and driven me to explore new space in our field and accomplish this thesis.

I am also grateful of the jury members for generously accepting to evaluate my thesis.

But most of all, I would like to thank and express my deepest gratitude to my family and friends for their continuous support and love especially my dearest Alireza who kept his amiable company with me in the ups and downs during my studies, the most wonderful parents whom I am eternally grateful and my beloved ones.

RÉSUMÉ

Les exigences de la connectivité mobile massive de différents appareils et de diverses applications déterminent les besoins des prochaines générations de technologies mobiles (5G) afin de surmonter les demandes futures. L'expansion significative de la connectivité et de la densité du trafic caractérisent les besoins de la cinquième génération de réseaux mobiles. Par conséquent, pour la 5G, il est nécessaire d'avoir une densité de connectivité beaucoup plus élevée et une plus grande portée de mobilité, un débit beaucoup plus élevé et une latence beaucoup plus faible.

En raison de l'exigence d'une connectivité massive, de nombreuses nouvelles technologies doivent être améliorées: le codage des canaux, la technique d'accès multiple, la modulation et la diversité, etc. Par conséquent, compte tenu de l'environnement 5G, surcoût de signalisation et de la latence devrait être pris en compte [1]. En outre, l'application de la virtualisation des accès sans fil (WAV) devrait également être considérée et, par conséquent, il est également nécessaire de concevoir la plate-forme matérielle prenant en charge les nouvelles normes pour la mise en œuvre des émetteurs-récepteurs virtuels.

L'une des nouvelles technologies possibles pour la 5G est l'accès multiple pour améliorer le débit. Par conséquent, au lieu d'OFDMA utilisé dans la norme LTE (4G), l'application d'une nouvelle technique d'accès multiple appelée Sparse Code Multiple Access (SCMA) est investiguée dans cette dissertation. SCMA est une nouvelle technique d'accès multiple non orthogonale du domaine fréquentiel proposée pour améliorer l'efficacité spectrale de l'accès radio sans fil [2]. L'encodage SCMA est l'un des algorithmes les plus simples dans les techniques d'accès multiple qui offre l'opportunité d'expérimenter des méthodes génériques de mise en œuvre. En outre, la nouvelle méthode d'accès multiple est supposée fournir un débit plus élevé. Le choix du codage SCMA avec moins de complexité pourrait être une approche appropriée. La cible fixée pour cette recherche était d'atteindre un débit d'encodage de plus de 1 Gbps pour le codeur SCMA.

Les implémentations de codage SCMA ont été effectuées à la fois en logiciel et en matériel pour permettre de les comparer. Les implémentations logicielles ont été développées avec le langage de programmation C. Parmi plusieurs conceptions, la performance a été améliorée en utilisant différentes méthodes pour augmenter le parallélisme, diminuer la complexité de calcul et par conséquent le temps de traitement. Les résultats de la mise en œuvre logicielle ont permis d'atteindre un débit de 3,59 Gbps, soit 3,5 fois plus que le débit cible.

Pour la mise en œuvre matérielle, une synthèse de haut niveau a été expérimentée. Pour ce faire, les fonctions et les bancs de test exprimés en langage C et développés pour les implémentations logicielles ont été utilisées comme entrées pour Vivado HLS. En ce qui concerne les caractéristiques de la conception réalisée avec Vivado HLS, cet outil est guidé par un ensemble de contraintes et directives appliquées aux modèles pour obtenir les meilleurs résultats. L'exploitation des contraintes et directives a permis d'obtenir des mises en œuvre performantes. Les meilleurs résultats de synthèse obtenus ont produit un encodage à 8 Gbps, soit 8 fois plus que le débit cible. En dépit du fait que la principale préoccupation de la mise en œuvre du codage SCMA est d'avoir un débit plus élevé, afin de déterminer les meilleurs résultats d'implémentations matérielles, La complexité de la zone a également été considérée. Par conséquent, la meilleure conception a été choisie en fonction de l'analyse de type Aire×Temps.

ABSTRACT

The demands of massive mobile connectivity of different devices and diverse applications at the same time set requirments for next generations of mobile technology (5G). The significant expansion of connectivity and traffic density characterize the requirements of fifth generation mobile. Therefore, in 5G, there is a need to have much higher connectivity density, higher mobility ranges, much higher throughput, and much lower latency.

In pursuance of the requirement of massive connectivity, numerous technologies must be improved: channel coding, multiple access technique, modulation and diversity, etc. For instance, with 5G, the cost of signaling overhead and latency should be taken into account [1]. Besides, applying wireless access virtualization (WAV) should be considered and there is also a need to have effective implementations supporting novel virtual transceiver.

One of the possible new technologies for 5G is exploiting multiple access techniques to improve throughput. Therefore, instead of OFDMA in LTE (4G), applying a new multiple access technique called Sparse Code Multiple Access (SCMA) is an approach considered in this dissertation. SCMA is a new frequency domain non-orthogonal multiple access technique proposed to improve spectral efficiency of wireless radio access [2]. SCMA encoding is one of the simplest multiple access technique that offers an opportunity to experiment generic implementation methods. In addition, the new multiple access method is supposed to provide higher throughput, thus choosing SCMA encoding with less complexity could be an appropriate approach. The target with SCMA was to achieve an encoding throughput of more that 1Gbps.

SCMA encoding implementations were done both in software and hardware to allow comparing them. The software implementations were developed with the C programing language. Among several designs, the performance was improved by using different methods to increase parallelism, decrease the computational complexity and consequently the processing time. The best achieved results with software implementations offer a 3.59 Gbps throughput, which is 3.5 times more that the target.

For hardware implementation, high level synthesis was experimented. In order to do that, the C based functions and testbenches which were developed for software implementations, were used as inputs to Vivado HLS. Regarding the characteristics of the design and Vivado HLS features, different sets of constraints and directives were applied to the designs to achieve the best

results. Finding the proper set of constraints and directives is quite challenging. The best achieved results with high level synthesis achieved an 8 Gbps throughput, which is 8 times more that the target. In spite of the fact that the main concern in SCMA encoding implementations is to have high throughput, to find the best design, the area complexity was also considered. Therefore, the best design was chosen based on an Area×Time analysis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

3GPP	3rd Generation Partnership Project
BDMA	Beam Division Multiple Access
CDMA	Code Division Multiple Access
CQI	Channel Quality Indication
CSM	Code-Space-Multiplexing
DVS	Distributed Virtual Switch
E2E	End to End
FBMC	Filter-Bank Multi-Carrier
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FTN	Faster Than Nyquist
GbE	Gigabit Ethernet
GFDM	Generalized Frequency Division Multiplexing
GPRS	General Packet Radio Services
IDMA	Interleave Division Multiple Access
IoT	Internet of Things
IPTV	Internet Protocol Television
LDS	Low Density Spreading
LTE	Long Term Evolution
M2M	Machine to Machine
MAC	Media Access Control
MIMO	Multiple-Input Multiple-Output
MPA	Message Passing Algorithm

MUD	Multi-User Detection
MVNO	Mobile Virtual Network Operator
NGMN	Next Generation Mobile Networks
NIC	(Network Interface Controller)
NOMA	Non-orthogonal Multiple Access
OFDMA	Orthogonal Frequency Division Multiple Access
OMA	Orthogonal Multiple Access
PHY	Physical
PIP	Physical Infrastructure Provider
PRE	Physical Resource
PTP	Precision Timing Protocol
RAT	Radio Access Technology
RDMA	Remote Direct Memory Access
RRC	Radio Resource Control
RTL	Register Transfer Level
SCMA	Sparse Code Multiple Access
SDN	Software Defined Networking
SIMD	Single Instruction Multiple Data
SMS	Short Message Service
SoC	Systems on Chip
SON	Self Organizing Networks
SP	Service Providers
TDMA	Time Division Multiple Access
UFMC	Universal Filter Multi-Carrier

VN	Virtual Network
VNO	Virtual Network Operators
VNP	Virtual Network Provider
VO	Virtual Operator
WAM	Wave Amplitude Modulation
WAV	Wireless Access Virtualization
WNC	Wireless Network Cloud

CHAPTER 1 INTRODUCTION

1.1 Concept Overview

Nowadays, the influence of information and communication technologies on the world economy cannot be denied and every innovation in this field is able to improve it. In the past, mobile communication was an expensive and luxury technology that just few people could afford, but today, it has become an accessible technology that is available almost everywhere and used by almost everyone. The most critical part in communication technologies is wireless communication networks that play an undeniable role in affordable broadband connectivity in the information society we have today. Also, it is growing extremely fast. For instance, mobile phone has changed from very simple devices with a simple screen and little processing capability to multitask devices with a large screen and a powerful processor. Besides this transformation in mobile phones, new wireless services such as social networking, web browsing and multimedia streaming are used more and more. In addition, users of other mobile devices like tablets have been added to mobile phone devices and constitute a large number of additional users of mobile devices. Furthermore, the mobile data traffic has been growing especially between 2010 and 2015 and it is expected to grow more in coming years, thus the current networks needs to be redesigned to increase coverage, data rate and capacity [1].

Obviously people can communicate more effectively in social or business manners when wireless networks develop more. Wireless networks have changed profoundly since the beginning of mobile communication systems. The network evolved from a simple telephone network which supports analog voice to a complex multitask network that supports hundreds of thousands of diverse applications and multimedia for billion users.

The first generation of mobile communication network was an analog radio system in 1980s but the first digital radio system that was called 2G introduced text messaging as Short Message Service (SMS) and circuit-switched data service with a maximum data rate of 9.6kbps. Later on, the General Packet Radio Services (GPRS) was offered in 2.5G, which introduced packet data in cellular networks. Then the first mobile network that was able to deliver data with higher bandwidth radio interface was unrolled as 3G that today is known as 3GPP (3rd Generation Partnership Project) and that is now used globally. The fourth generation of mobile network (4G) is LTE (Long Term

Evolution) or LTE-Advanced that is a more advanced release of more or less the same technology. The deployment of 4G happened in coincidence with a significant evolution of telecommunications systems supporting networks and mobile devices. [2] [3]

Although packet-switched service was introduced in 3G, it became a very important design goal for 4G as well as IP services. This service could provide a wide range of different services along with different requirements. The requirements include high data rate, high capacity and low latency targets. The maximum data rate in 2G was on the order of kbps, in 3G it was on the order of Mbps but for 4G it should have been close to Gbps. Capacity to provide data rate for lots of users was another challenge in 4G, because the shortage in capacity influences the quality of service. Capacity has been measured as spectral efficiency. In 4G, latency plays a critical role in real time applications, and it was measured as a delay experience by a packet from a server to its reception by a user. [3]

3G and 4G networks were mainly designed to provide a consistent coverage, but that coverage was specifically intended for outdoor services, particularly when it is delivered with macro-cells. By contrast, the upcoming fifth generation mobile technology (called 5G) is a heterogonous frameworks. High data rate services are offered by leveraging indoor hotspots, and voice services as well as other data rate services are always consistent in the whole system. These kinds of services make differentiate 5G. Also, in 5G users should be able to have both indoor and outdoor connectivity simultaneously. Therefore, one of the main changes in 5G should be a design combining macro based telecommunication paths and local telecommunication paths [4].

Form another point of view, Internet of Things (IoT) is becoming more and more popular, and it can be highly beneficial if it is utilized for industrial purposes or in health related applications. This should increase in amount of data that the network must handle, setting a requirement for extra capacity that requires both additional spectrum as well as higher spectral efficiency.

In order to provid service to many users, bandwidth must be expanded. Although 4G supports high downlink data rates of up to 1 Gbps, 5G networks need to provide much higher data rate and extended coverage. Therefore, lots of interests go to beyond 4G technologies. The most important and the first thing that should be reconsidered for next generation of wireless networks is wireless standards, specially data rates and spectral efficiency.

The prediction says that fifth generation wireless networks will be deployed around 2020. There are clear requirements for low latency and more than one Gbps data rates [5]. In order to improve spectral efficiency, a main challenge is to have smaller cells for network nodes, better interference mitigation and using massive MIMO (multiple-input multiple-output) techniques.

Furthermore, different wireless devices have different variety of features and characteristics such as cost, hardware platform and processing power. Accordingly, different applications need different and diverse data rate and latency. Therefore, diversity in devices as well as in the applications should be taken into account as another technical challenge.

As a brief, some of the main 5G requirements in comparison with 4G discussed in the NGMN (Next Generation Mobile Networks) white paper [6] are:

- Higher capacity, 100 to 1000 times more
- Providing Gbps data rate everywhere
- Less that 1ms latency for E2E use case
- Higher connectivity rate, 10 to 100 times more

By providing these main requirements for 5G networks, users will experience better quality of services in terms of connectivity and data rate. To deliver higher data rate and capacity, some technologies should be considered such as massive MIMO, millimeter wave spectrum usage, D2D (device to device) connectivity and using more small base station in order to support multi-RAT (Radio Access Technology). Using small base stations can leverage small cell networking to enhance inter-cell interference but it needs to be improved in terms of its performance in the peak traffic. [7]

The predictions express that in the next decade the data rate will be doubled each year and to prevent increasing in the users' costs, some new, innovative and cost effective wireless access should be provided by the wireless communication industries. In this way, to achieve new applications like machine to machine (M2M) communication and cloud computing require diverse features. Also, advanced signal processing, heterogeneous base stations and finding new way to use spectrum much more efficient can be some of potential technologies for 5G.

Furthermore, being continuesly connected has more benefits for people that just being able to connect in the future and even now that new generation of wireless networks are trying to

provide. These benefits include diverse applications to control the traffic, monitor different systems for safety, having remote access to the household appliances, medical purposes and much more. As a consequence, for making next generation of mobile networks commercialized and available for users the preliminary step is to figure out the requires characteristics and features. The second step is to reconsider the existing standards for wireless networks and prepare new standard package for components of 5G system.

However, data traffic demand from mobile board band will increase more in the future. With this assumption that the wireless networks deliver the service to the users with the same energy consumption like existing networks, the cost of delivering a bit to the end user will significantly increase in the massive traffic. Therefore, 5G networks have to offer lower energy consumption components to keep the cost of traffic as low as possible. In addition, for 5G networks there is a need to provide data rate about 10Gbps for specific purpose but the data rate that should generally be accessible for the users must be more that 100Mbps. Finally at least a few Mbps must be available everywhere and all the time. In addition to high demand for traffic and data rate, machine type communication like M2M needs some other requirements to be used in 5G [7].

The applications related to machine type communication needs very low access latency to perform their tasks perfectly, specially in critical mission such as traffic safety or some emerging industrial Internet applications that need less that 1ms as latency. LTE provides acceptable latency for lots of applications but for latency-critical applications this amount of latency is not acceptable to be sure that the application works well in order to perform its task in a timely manner. Also, this kind of applications requires a reliable connectivity that the network is able to guarantee its availability. And this reliability for the service should be higher than what exist in LTE as well which has already had a high reliability.

On the other hand, as it is mentioned, 5G needs to deliver very low cost connectivity and traffic. Therefore, for massive machine type connectivity 5G can use very large number of connected devices and sensors with low cost connections that consume a small amount of energy. Thus they can operate continuously for several years without need to recharge. Consequently, some features, performances and capabilities of next generation of wireless access needs to be improved. These improvements should be done in dimensions such as traffic capacity, higher data rate, access latencies and massive diverse connectivity that are figured out from the user experience of LTE

wireless networks.

In brief, beyond 2020 5G technology will provide a telecommunication service to deliver information and data to anyone and anything, anytime and anywhere in land.

1.2 Research Objectives

In comparison with what today wireless networks deliver, next generation must provide a much wider range of access requirements. In order to do that, 5G needs to consider a novel multiple access technique which has a high spectral efficiency. Besides, the performance of wireless communication networks depends on multiple access techniques. Thus, multiple access techniques always are the key technology in wireless network from the first generation till today. In general, there are two classes of multiple access techniques that are orthogonal and non-orthogonal. This classification is done based on how the resources are dedicated to the users. Since in the orthogonal multiple access techniques (OMA), no inter-user interference exists, the receiver can utilize a low complexity detection method to receive the signals form users. However, in the non-orthogonal multiple access techniques (NOMA), there is interference among users because all users can use all resources simultaneously. Thus, there is a need to use more complex techniques in the receiver that are called as multi-user detection techniques (MUD) [8]. In the OMA techniques, signals from users are placed orthogonally to each other and therefore the cross correlation between them are zero. While in NOMA techniques, inter-cell interference become important and they have non-zero cross correlation.

Both OMA and NOMA techniques have their own advantages and disadvantages. OMA cannot provide a high spectral efficiency in uplink and system upper bound, while NOMA techniques deliver high spectral efficiency. On the other hand, OMA techniques have a proper performance but NOMA techniques needs complex MUD techniques that must be implemented at user devices and because of limited processing capability in users' equipment, that is not easy. Since the spectral efficiency plays an important role in delivering service to users in a fair manner in the system and considering a high demand for it in the future, NOMA technique can be a better option than OMA techniques. In an optimal NOMA techniques, users share the resources in time or frequency domain but the number of users of each subcarrier are not controllable that causes MUD technique implementation too difficult. However, some techniques such as CDMA (Code

Division Multiple Access), IDMA (Interleave Division Multiple Access) and LDS (Low Density Spreading) by using some coding and spreading methods can help NOMA to separate users at receiver [8].

In the first and second generations, FDMA (Frequency Division Multiple Access) and TDMA (Time Division Multiple Access) were used respectively. In 3G and 4G, CDMA and OFDMA (Orthogonal Frequency Division Multiple Access) have been used. The previous orthogonal techniques users are allocated orthogonally to the resources and the resources could be in time, frequency or code domain, while NOMA techniques are usually based on power domain instead of time, frequency or code domains and make all subcarriers available for each user. Although these orthogonal multiple access techniques have provided acceptable gains, for 5G considering spectral efficiency and Internet of Things (IoT) requirements, NOMA techniques are better choice. In addition, NOMA can provide the timely manner service to users with different channel conditions that eventuates very low latency and very high connectivity [3].

The first generation of mobile networks delivered a data rate up to 2.4kbps while the second generation provide the data rate of 64kbps to 144kbps. The 3G networks in the beginning has data rate of 2Mbps but after some improvement in this generation the delivered data rate was increased about 5Mbps to 30Mbps. Although in 4G, data rate has been improved up to 1Gpbs that is very significant in comparison with previous generations, regarding the high demand for connectivity in 5G this amount is not enough [4].

In 4G system, an orthogonal frequency division multiplexing (OFDM) used with an advanced radio interface. This system provide up to 1Gbps. OFDM can be used as multiple access technique that provide separated transmission in frequency domain to or from different terminals. In uplink, for data transmission from different terminals a set of subcarriers are used and also in downlink for data transmission to terminal a set of available subcarrier are used. This process is called OFDMA (Orthogonal Frequency Division Multiple Access). Since in OFDMA transmission is done from different terminals, time alignment in the base station is very important. Therefore, there is a need to use some timing and synchronising control techniques. Even with a perfect transmission time control scheme, interference between some subcarriers is inevitable because of some frequency errors. While in a NOMA scheme there is no need to have such synchronization [3].

CDMA is an orthogonal approach which is used in 3G networks has a proper performance in terms of inter-cell interference cancelling and robustness against fading but it is not suitable for high data rate especially for asynchronous transmission environment. Consequently, this technique is not a good choice for 5G and it is needed to utilize other approaches [9].

Another approach for 5G multiple access is combining two techniques, one orthogonal and one non-orthogonal. The idea is to consider two regions for the cellular network and use one technique for inner region and the other technique for outer region. OFDMA can be used for inner region and a specific case of CDMA (CP-CS-CDMA) for outer region. The reason of using OFDMA for inner region is the interference and the reason of using CDMA for outer region is higher system capacity. It seems that to choose between a NOMA scheme and a combination scheme for 5G networks, it is required to determine the radio technology aspect for 5G in order to adapt an appropriate multiple access technique to that [9].

In almost all the present multiple access techniques, the resources such as time and frequency are divided between the users, therefore each user has a portion of resources and the system capacity depends on time and frequency that make some limitation for the system capacity. Another proposed multiple access technique for 5G is BDMA (Beam Division Multiple Access) that is independent on time or frequency. In BDMA, the antenna beam is divided by the base station and dedicated to mobile station according to mobile station location. This scheme helps to increase the system capacity. Since, the angle of mobile stations to base station can be different, base station can transfer data to different mobile stations simultaneously. BDMA is expected to provide more than 1Gbps that make it a very interesting option for 5G but its performance can be challenged when some mobile stations locations are in the same angle with base station. In the case the mobile stations are in this situation, they need to share the same beam thus the data rate they receive is less than the case the mobile stations are in different angle with base station. Also the beam these mobile stations share should have lower peak to decrease power ratio problem [10].

The 5G mobile technology is characterized by a tremendous growth in connectivity and density of traffic. In comparison with 4G-LTE (Long Term Evolution), fifth generation wireless technology aims to support massive connectivity of different devices and diverse applications at the same time, with levels of performance beyond what is offered by LTE. In pursuance of the requirement of massive connectivity, numerous new technologies are to be improved: channel

coding, multiple access, modulation and diversity etc. Therefore, the cost of signaling overhead and latency should be taken into account while considering the 5G environment [11].

CDMA technique works like a coding procedure that encode the binary data to a multidimensional complex data. LDS is a specific scheme of CDMA that use a few nonzero elements in coding sequence. SCMA (Sparse Code Multiple Access) have some similarity with CDMA and specially LDS but it is a non-orthogonal multiple access. SCMA uses coding process form binary domain to multidimensional complex domain. This encoding process can be done differently for different users and in the receiver users' data can be detected using MPA (Message Passing Algorithm) because of sparse feature of coding process. By using SCMA, system can be overloaded like LDS when the number of users are higher than spreading factor which is the codeword length in SCMA [11] [12].

As LTE networks do not support massive connectivity in the uplink, SCMA as a new multiple access scheme with the characteristics of providing massive connectivity can be a proper substitute for OFDMA. SCMA with a large number of layers allows system to be overloaded that enables massive connectivity. Each layer in SCMA represent a user and has a specific coding process as a codebook.

The research objective in this project is to implement SCMA main uplink algorithm, as a new multiple access technique, SCMA encoder to propose a new encoding system for next generation of mobile networks.

1.3 Contribution

As it was explained in previous section, one the basic requirements for 5G is new multiple access and a NOMA technique is much more suitable for this purpose. SCMA is one of the most novel multiple access techniques that is proposed for 5G. It is expected to provide more than 1Gbps data rate and because of sparsity it is also expected that the implementation complexity of SCMA algorithms are in the acceptable levels. Therefore, in this thesis, the efforts are focused on implementing one of this algorithms that is fundamental for uplink SCMA system. This algorithm includes SCMA encoding part and multiplexing part that have the responsibility of encoding binary data of SCMA layers to multidimensional complex codes and provide multiplexing for physical resources.

My main contribution in this project is to implement a SCMA encoding system both in software and in hardware in order to have more than 1 Gbps throughput. Another contribution is to experiment high level synthesis as a proper substitude for low level synthesis and HDL codes. The results of this experiment are useful not only in implementing SCMA encoding but also in implementing other algorithms needed for 5G wireless communication.

In the process of SCMA encoding implementation, since the main application for uplink SCMA system is for base stations, two types of implementation are considered with a goal of delivering a data rate more than 1Gbps. In order to do that the processing time for each bit of each data layer must be less than 1ns. Therefore, encoding algorithm should be deeply reviewed in terms of computational complexity to figure out the obstacles in minimizing processing time for a bit. The implementations both in software and hardware are done and have being improved regarding algorithm complexity assessment. Sometimes, to reduce computational complexity the algorithm and consequently the implementation code needs to be re-arranged but these re-arrangements and other complexity reduction techniques almost never reduced the accuracy of encoding.

Although the most popular way to implement a hardware on FPGA is using HDL codes, in this thesis another way has been experimented that is High Level Synthesis (HLS). For HLS instead of a low level programming language like HDL, a high level language like C, C++ or SystemC is used that helps to improve time and cost efficiency of hardware implementation. The chosen language for software implementation in this thesis is C that make a proper opportunity to re-use the codes for HLS implementation as well.

The achieved results were benchmarked using 4G data rate range and the turbo encoder implemented by Xilinx. These results illustrated that the processing time for each bit in the implemented SCMA encoding is much less that 1ns both in software and hardware. Therefore, the achieved data rate is much more than 1Gbps that 4G delivers in uplink. Consequently, the target dedicated to this project was obtained.

1.4 Dissertation Plan

This dissertation choses SCMA encoding technique to experiment implementations for a possible uplink system in 5G networks and proposes two types of SCMA encoding implementations, software and HLS. The organization of this document is as follow. Chapter 2

reviews the literatures considering Virtualization in wireless networks, 5G networks, SCMA technique and implementation methodology in 4 sections. Chapter 3 explains the SCMA encoding algorithm and implementations experiments. In this chapter, the improvement process and experiments performed to achieve the target in terms of high data rate is discussed. In addition, it includes explanation about how to use different options of hardware synthesis. This describes all experiments done in Xilinx Vivado HLS. The results of all different implementations both in software and hardware are listed in Chapter 4. This chapter provides an opportunity of making comparisons among software and hardware implementations regardless of type of implementation and also between different versions of software implementations and hardware implementations themselves. The achieved results are calculated for SCMA encoding system contain SCMA encoder and physical resource multiplexing. All these results are studied and benchmarked in order to validate them and represent the high performance SCMA encoding system.

CHAPTER 2 EFFICIENT HARDWARE ARCHITECHTURE FOR NEXT GENERATION OF MOBILE NETWORKS

In homogeneous or heterogeneous wireless virtual networks (VNs), hardware platform architectures need to assign resources to perform different functionalities in order to support Wireless Access Virtualization (WAV), which is a major challenge. Therefore, new accelerator architectures are required to enable WAV at both PHY (Physical) and MAC (Media Access Control) layers. New architectures should consider deterministic low latency means for computation that could have energy efficiency management scheme as well. Co-operation between multi-tier Heterogenous Networks (HetNets) with different parameter settings and timing synchronization schemes is another challenge of future 5G systems. In addition, some issues that should be considered in the design of baseband accelerators are listed as follows:

- Defining efficient hardware platform architectures to enable WAV using CPU cores, DSP cores and baseband accelerators;
- Forward error correction (FEC) encoding and decoding;
- MIMO signal processing;
- Developing virtual cores to support new features such as channel aggregation and distributed signal processing across distributed platforms;
- Designing baseband accelerators with the least-overhead that offer distributed signal combining and interference suppression techniques;
- Considering various constraints related to synchronization complexity, throughput and latency in baseband accelerators design;
- Implementing synchronization mechanisms in the HetNet WAV.

Considering that one of the main challenges in this thesis is reducing computation time in the proposed architecture in next generation of wireless networks (5G), several previous contributions were reviewed to leverage methods and protocols for 5G that can lead the project to define a high performance hardware architecture. Some results of this literature review are summarized in this chapter that is divided into six sections. In Sections 2.1, 2.2 and 2.3, the literature concerning wireless network virtualization, 5G networks and research ideas about 5G are reviewed respectively. In Section 2.4, SCMA algorithms are reviewed. Section 2.5 describes generic tools and methodology for hardware implementation. Finally, Section 6 summarizes the key points

reviewed in this chapter.

2.1 Wireless Network Virtualization Literature

Virtualization in wireless networks allows a node to split physical resources between different system users [13]. Although many technical issues must be addressed for successful realization of virtualization, flexibility, security, diversity and manageability are provided for multiple heterogeneous network architectures sharing a substrate of network virtualization [14], [15]. Extra functionality at no extra cost and better resource control are the most important potential benefits of virtualization [13]. The first definition of network virtualization categorizes the roles of the traditional Internet service providers into two independent entities. These two entities are infrastructure providers and service providers, which manage physical infrastructure and create virtual networks respectively [16], [17].

The first step toward virtualization is providing isolation between multiple virtual entities but, in the wireless context, isolation is not guaranteed by over provisioning [13]. Also, as wireless spectrum does not increase, advanced resource isolation models are needed. The second step involves resource partitioning, which may depend on the hardware capabilities. Thirdly, wireless technologies enable a different level of flexibility in the medium access operations, which does not generally allow full control of the medium and differentiation of virtual entity behavior [13].

There are different levels of virtualization of wireless networks resources based on levels of isolation, flexibility and partitioning. A temporal partitioning of the hardware in terms of channel switching and power saving, and a MAC scheme for virtual interfaces are the most common solution for virtualization. In addition, solutions for virtualization can be a simple scheduling virtualization framework or a complex low level differentiated MAC functionality management [13].

The idea proposed in [13] is performing virtualization functions utilizing a hardware platform in terms of a MAC engine, which is abstracting the device resources and capabilities, and a virtual MAC monitor, which is solving the hardware conflicts. The MAC machine, which is proposed in [6], is a MAC program executer. It is a high-level state machine that can be implemented on different system cards as follows [13]:

- A source state, which is the starting state of state machine operations;

- A trigger event, like a signal generated by the hardware platform;
- An action, representing an atomic program code, which can also work on the hardware configuration registers;
- An optional guard condition, to be evaluated after the trigger event.

The responsibility of the virtual MAC monitor, which is proposed in [13], is controlling and managing access to the hardware. To implement a virtual MAC monitor, there are two approaches. The first approach is a classical time-based mechanism. This mechanism assigns the hardware to a single virtual entity in a given time interval. The second approach is a virtual collision management mechanism. This mechanism uses time slicing and virtual slicing to perform multiple accesses to the hardware. In the time slicing approach, the virtual MAC monitor enables the MAC engine to do thread switching and in case of virtual collision, the virtual MAC monitor enables parallel MAC machine execution.

Also in [13], a MAC virtualization architecture is proposed, whose main components are the MAC Engine and the Virtual MAC Monitor, where the MAC Engine acts as a host. It is an abstraction of the hardware in terms of actions, event signals and configuration. The Virtual MAC Monitor allows exposing multiple virtual engines, which can independently run their guest MAC machines and corresponding upper network applications.

In [15], an architecture for network virtualization is proposed. It is composed of four main parts: Physical Infrastructure Providers (PIPs), Virtual Network Providers (VNPs), Virtual Network Operators (VNOs) and Service Providers (SPs). In this architecture, the SP gives his requirements to the VNO. This VNO encloses its needs on the VNet, and then on a chosen VNP, which is responsible for assembling the VNet, is provided with this description. Each VNet has a data plane, which refers to a virtual network context, and a control plane, which is necessary for specific VNet management tasks during VNet operation. This architecture defines the interaction between the roles of this architecture without prescribing their internal organization, structure and policies.

Generally, virtualization of the wireless medium is performed based on a Time Division Multiple Access (TDMA) virtualization technique [19]. The main purposes of wireless medium virtualization are sharing the network infrastructure and sharing an over-provisioned wireless infrastructure with lower usage of resources [20], [21]. Although sharing the use of wireless

medium bandwidth is a form of wireless medium virtualization, the main goal of this sharing is maximizing the use of the wireless medium while preserving the quality of service.

In [19], a wireless network virtualization method is proposed. That method considers network usage and quality of service. The method introduces a scheduler to organize wireless medium access from a node. Inside the node, there is a simple traffic management aspect. It stores the arriving packets in a virtual operator (VO) queue and then the TMDA scheduler sends the corresponding packets to the wireless interface based on the type of traffic and scheduling techniques.

A packet that enters the VO queues suffers from a delay until it is transferred to the wireless interface; that is called packet delay. The duration of this delay depends on the number of packets waiting in the VO queue, the time slot duration, the number of VOs and the available bandwidth. The variation in delay, especially in services like voice application, is called jitter [22]. If the VOs have only one traffic, the delay is very low, but if they have two traffics the delay stays very similar to their peers in the symmetric scenarios.

In [23], the authors propose Cabernet (Connectivity Architecture for Better Network Services), a three-layer network architecture that lowers the barrier for deploying wide-area services. They introduce many challenges: how do the connectivity providers build virtual networks, and what do they need from the infrastructure providers? What is the functionality required at the infrastructure routers or servers to realize Cabernet while achieving high performance? How can the network services run on this layered architecture? The paper discusses how these challenges are addressed in the Cabernet design. Although IPTV (Internet Protocol Television) delivery is the case study used to develop and explain this design, there is no experimental result in this paper.

In [24], the authors propose a wireless network cloud (WNC) for a wireless access network. It provides all the necessary transmission and processing resources in a cloud. They analyze several important system challenges involving computational requirements of virtual base stations, I/O throughput, and timing networks for synchronization.

They consider six scenarios to implement WNC (just description, no details or results):

1) considering WNC as a resource pool to support various kinds of wireless access networks,

2) considering a Mobile Virtual Network Operator (MVNO),
3) considering different wireless traffic in the daytime and nighttime,
4) involving the rural area,
5) collaborating of multiple BSs (Base Station) in the PHY layer signal processing such as in cooperative MIMO, and
6) using open IT platforms in the WNC.

They mention that precise synchronization and timing is required to avoid interference. The IEEE 1588 Precision Timing Protocol (PTP) is used in this platform. Three kinds of structure are considered to construct the virtual BS pool. In the first structure, the software packages of virtual BS-PHY and virtual BS-MAC are combined into one virtual BS. In the second one, virtual BS-PHY and virtual BS-MAC are separated. Furthermore, one virtual BS-MAC can serve multiple virtual BS- PHY components. The third structure supports the cooperative radio processing which implies splitting the workload between base stations for most efficiency.

Reference [24] explains that multithreaded processors such as IBM wire-speed processor (with 16 cores and 64 hardware threads) and the Raza XLR processor (with 8 cores and 32 hardware threads) could match the multithreaded program model in the MAC layer. Also, the authors propose network accelerators and cryptographic coprocessors to accelerate MAC-layer processing. Furthermore, they state that the MAC-layer requires 10% of the computation resources of the whole BS, while the PHY layer uses 90% of the resources. It is stated in this paper that only 29.6% of the instructions in the PHY layer are SIMD instructions.

In [24], system throughput of the WiMAX BS-PHY stack using a Cell/B.E. Blade is demonstrated for three scenarios. The first scenario is related to computation only. In this case, all the R-B link data are stored in memory rather than in the Ethernet interface. The R-B link refers to the link between the radio front end and the virtual BS pool. This link is able to support the topology of multiple-point to multiple-point models. The second result is for the scenario with computation and data transfer over a 10GbE (Gigabit Ethernet) interface without RDMA (remote direct memory access). Compared to the case of computation-only, it has a 68% performance loss. The third result is for the scenario with computation and data transfer over a 10GbE interface with remote direct memory access (RDMA). It only has a 4.5% performance loss when compared with the one for computation-only. With RDMA, data can directly be moved between the memory of the RRH and

the virtual BS, or between two virtual BSs, without involving the operating system of either one. Only a small overhead is required for a zero-copy protocol used in RDMA.

The IEEE 1588 PTP provides a means by which networked computer systems can agree on a master clock reference time, as well as a means by which slave clocks can estimate their offset from a master clock time. The closer the time stamp is taken from the hardware transmission or receipt of the messages, the smaller the latency, and consequently, the accuracy is also better (of the order of 10μs).

As a conclusion, the WNC proposes two important ideas that can be useful for new wireless architectures. First, it adopts an open IT architecture that could replace today's proprietary hardware design in a BS system. Second, cloud-computing concepts are used in building the wireless access network. To meet the computational requirements, SIMD techniques or reconfigurable hardware accelerators (e.g., FPGAs) of channel decoders can be considered in future multicore and SMT-based system designs. To relieve the CPU overhead for the high I/O throughput on an R-B link, different methods should be considered for the virtual BS platform, such as RDMA over the R-B link, processors with network accelerators, or the advanced Ethernet NIC (Network Interface Controller) with an OS-bypass functionality. To construct an accurate timing network with the IEEE 1588 PTP, the authors in [24] propose a software implementation using multicore processors with a network accelerator.

The challenge in [25] is to allow multiple OS images to transparently share the same physical server and I/O devices. It requires supporting local switching between different virtual machines within the same server. In [25], Cisco and VMware have collaborated to define a set of APIs that enable transparent integration of third-party networking capabilities within the VMware Virtual Infrastructure. VN-Link (Virtual Network-Link) provides another solution as a Cisco Distributed Virtual Switch (DVS) running entirely in software within the hypervisor layer (Cisco Nexus 1000V Series).

In [26], the challenge is to analyze the feedback mechanisms for CoMP transmission modes. The aim is reducing the overhead of signaling in CoMP. Authors in this paper mention implicit feedback methods for different CoMP modes. They have two proposals. Firstly, they recommend per cell reporting of the feedback regardless if an implicit or explicit one is used. Secondly, they recommend if implicit feedback is agreed upon mechanism, a single additional joint

Channel Quality Indication (CQI) is included along with the per cell feedback.

In [27], the idea is to deploy and evaluate researchers' ideas with real routing software, traffic loads, and network events. The challenge is to explore a set of concepts and techniques, which facilitate flexible, realistic, and controlled experimentation (e.g., multiple topologies and the ability to tweak routing algorithms) on a fixed physical infrastructure. Virtual network infrastructure's (VINI's) high level design and the challenges of virtualizing a single network are presented. Further, PL-VINI, an implementation of VINI on PlanetLab, running the "Internet In a Slice" is proposed. The evaluation of PL-VINI shows that it provides a realistic and controlled environment for evaluating new protocols and services.

Based on this paper, constructing a virtual network involves solving four main problems. First, the infrastructure must provide support for virtualizing network devices and attachment points because a network researcher may wish to use the physical infrastructure to build an arbitrary topology. Second, once the basic topology is established, the infrastructure must facilitate running routing protocols over this virtual topology. Third, once the virtual network can establish its own routing and forwarding tables, it must be able to transport traffic to and from real networks. Finally, the virtual network infrastructure should allow multiple network researchers to perform the above three steps using the same physical infrastructure. Experimental results show that running Internet In a Slice (IIAS) architecture on PL-VINI provides a 4 times increase in throughput and reduces variability by over 80%.

Authors in [28] provide some qualitative analyses on the principle of CoMP, and point out some issues within the scope of the CoMP technology. For intra-eNodeB co-operation, it is necessary to further investigate coordinated scheduling, channel measurement and estimation, interference management, and overhead. For inter-eNodeB cooperation, it is important to consider the balance between the flexibility of cooperation and the modification of X2 interface. Additionally, a virtual BS pool will require real-time support from the OS, hypervisor, and scheduler.

2.2 Next Generation Wireless Networks (5G) Literature

The fifth generation of mobile technology (5G), which is positioned to address the demands and business contexts of 2020 and beyond, is a main concern of our research. For the first step,

recognizing the status of 5G seems essential to figure out the up-link/down-link processing chains and suitable algorithms for each step. Thus, we performed a literature review that led us to believe that the 5G definition is in a very preliminary stage. In order to confirm this, we looked at the recently published NGMN (Next Generation Mobile Networks) White Paper that gives critical information that reviews features of 5G that are defined together with other aspects that are hard requirements or nice to have features. This is partly defined in comparison with 4G [6].

For future mobile communication networks, some technologies need innovation such as signal processing techniques, spectrum usage concepts, air interface numerology and usage of more heterogeneous base stations. Some key challenges for this purpose, which are inherent in designing and systems operation are interference, mobility and session management, and network infrastructure [18]. Regarding those key challenges, Cellular M2M (Machine-to-Machine) connections, Cloud computing and Three-dimensional (3-D) video are required for future mobile communication networks. Also, future systems will require more flexibility and especially for M2M connections, systems need to be more robust in time and frequency, while enabling low-power and low-cost operation in sensor nodes.

As interference scenarios are frequently changing, more complex, out of operators control and less structured, interference management is a key concern. Various solutions should be analyzed in consideration of required resources and various modulation schemes exploited for different services. Additionally, a fundamental question regarding interference management is where knowledge about interference is available and where decisions on its management are performed.

To answer this question two solutions are proposed: single-cell solutions and multi-cell solutions (decentralized solutions, centralized solutions). Compared to 4G, various use cases for 5G are associated with more handover between base stations. Thus, mobility and session management has to be performed more often based on radio condition, service availability for certain types of traffic and QoS level [18]. Some networks have features to enhance mobility, such as the ability to obtain and distribute information on the access point topography and traffic flow statistics, and ability to decide which cell should give service to a particular device and trigger handover. In future mobile communication networks, in order to simplify device architecture and reduce power consumption, device related overhead has to be minimized.

Another important challenge in future mobile communication networks is the way data is transmitted or information is exchanged between access points [18]. Considering required research, there is a need to investigate where devices have links to different cells in uplink and downlink connections or through data and control channels, in order to provide alternative connectivity and signaling concepts, respectively. Also, research should be performed on virtual cell concepts, alternative session management methods, and protocol stack modifications allowing reduced signaling overhead in low payload M2M connections.

Requirements for 5G can be defined relative to the level of performance in 4G. Thus, it is expected that 5G will provide: 100-1000 times higher system capacity, user data rates in the order of Gbps everywhere, latency in the order of 1 millisecond for E2E (End to End) use case, while supporting 10-100 times more connected devices per area, and offering 10 times longer battery life for portable devices [6], [7]. Whether all these characteristics will be met simultaneously remains to be seen.

To enable technologies and architectures for 5G wireless communication networks, there are some candidate solutions including: utilizing more compressive heterogeneous networks with large number of small base stations supporting various Radio Access Technologies (RATs), using very large Multiple Input Multiple Output (MIMO) arrays, utilizing millimeter wave spectrum where larger wider frequency bands are available, direct device to device (D2D) communication, and simultaneous transmission and reception, between others [7]. Hence, densification using multi-RAT HetNet can provide significant gains in average cell and celle-dge throughput possible in comparison with LTE-only small cells in a typical deployment with 4 outdoor small cells (pico cells) per macro cell [29]. Also, to achieve higher spectral efficiency for cellular systems, advanced MIMO techniques and specially MultiUser MIMO offers higher multiplexing gain. Direct device to device (D2D) communication are considered to support data exchange between user devices without using base stations or the core network. D2D could increase network capacity by reusing the spectrum, which is an effect similar to that obtained with macro cells. That could also be done by using otherwise unused unlicensed spectrum. In addition, full duplex would allow a wireless device to transmit and receive data in the same frequency band simultaneously. That would increase the physical layer capacity, which could be a good technology for 5G wireless networks [7].

In [30], a proposed cellular architecture would separate indoor and outdoor scenarios. In this architecture, a distributed antenna system and massive MIMO technology are required [31]. According to that proposal, a large number of antenna arrays are distributed geographically. Although this architecture would likely increase the cost of the infrastructure, it is expected that it would improve cell average throughput, spectral efficiency, energy efficiency and data rate in the long term. A massive MIMO system, which consists of multiple antennas at the transmitter and receiver, can enhance both spectral and energy efficiency [32]. As a novel MIMO technique, spatial modulation (SM) is proposed. In SM, some part of the data encodes information to be transmitted onto the spatial position of each transmit antenna in the array. SM can alleviate three important problems in MIMO systems: inter-channel interference, inter-antenna synchronization and multiple RF chains.

Cognitive Radio (CR) is another part of the architecture proposed in [30]. It is a software defined radio technique proposed to improve the utilization of the congested RF spectrum. Generally, a CR network needs to be aware of the surrounding radio environment and manage its transmission accordingly. As a result, the proposed techniques provide high quality and high data rate services to indoor users and reduce the pressure on outdoor base stations at the same time [33].

2.3 Promising Research Ideas

In order to propose enhancements contributing to the deployment of fifth generation mobile technology (5G), one of first steps is to understand the status of 5G. It seems essential to define some up-link/down-link processing chain, including algorithms suitable for each step. In order to confirm this observation, the NGMN White Paper [6] provides very useful information.

NGMN (Next Generation Mobile Networks) is an industrial cooperative initiative that has a core mandate of defining operator requirements for 5G. The NGMN White Paper was published in February 2015 as a guideline to facilitate 5G definition and design. The NGMN requirements discussed in the White paper cover a number of areas to leverage the overall success of LTE and the structural separation of hardware and software based on design principles and proposed architecture. This White Paper has been endorsed by the following NGMN Board Members: AT&T, Bell, BT (British multinational Telecommunications services Company), T (Deutsche Telekom), Kpn (Netherland Mobile Telecommunication Company), Kt (Korea Telecom

Company), NTT Docomo (Japan Mobile phone Operator Company), Orange, Singtel, Sk Telecom, TELE2, TELECOM Italia, Telefunica, TELEKOM Austria Group, Teliasorena, Telstar, TELUS, TURKCELL, VimpleCom and Vodafone [6].

According to the NGMN White paper, in its Appendix C, useful guidance regarding target applications is found. Various suggestions relate to energy consumption and power efficiency, and the following subjects could be related to our research [6]:

1) New waveforms;
2) Advanced multiple access technologies
3) Massive MIMO and enhanced multi-antenna schemes;
4) Interference coordination;
5) Technologies for small packet transmission;
6) Densification: Small cells/Ultra-dense networks;
7) Dual connectivity-capacity/coverage split system design;
8) Device-to-device communication;
9) Software-Define-Networking;
10) Virtualized mobile core network;
11) Virtualized C-RAN;
12) Micro-servers;
13) Intelligent heterogeneous management;
14) Embedded measurement of network performance;
15) All optical transport networks with optical router/switch.

From the 15 listed challenges, five topics will be described in more detail in the following paragraphs in order to provide a short explanation and review some possible solutions.

2.3.1 Technologies for small packet transmission

There are different schemes for small packet transmission with different Quality of Experience, but the three most important ones are:

1) Periodical keep-alive packets, which are transmitted with minimal non-intrusive content

One of the most important roles of keep-alive packets is verifying the status of the computer

at the remote end of a connection and determine whether it is still alive. Each keep-alive packet can be sent over a connection at every specific time slot called the KeepAliveTime, if there is not another data or higher-level keep-alive packet carried by the connection. When a keep-alive packet is not responded in another specific time slot, the KeepAliveInterval, it will be repeated. For instance, in TCP, an ACK message with sequence number for the connection, which should set to one less than the current sequence number, is a keep-alive packet. When a host receives an ACK, it will respond with another ACK with the current sequence number [34].

2) Bursty Instant Messages

As a comparison between voice and instant messaging, voice is a continuous stream of packets that can tolerate bit errors while instant messaging is bursty and does not tolerate errors, so instant messaging typically causes cycles of abstruseness and repair. Also, since it is asynchronous, short messages are sent in bursty fragments by the sources. For mastering issues that situation needs immediate action, and sources may send requests in parallel in the medium. Increased clarity is a result of using bursty instant messages [35].

3) Real-time critical message delivery

When a failure happens in distributed real-time systems, it is essential to maintain reliable and timely message delivery between nodes. It is needed that the system has the ability to deliver a message within its deadline, so the system has to perform a recovery action. Usually this recovery action implies additional costs to the system, which can be very high. These costs can be very high if the recovery action fails because of lack of time or resources [36].

Because of the features of the above small packet transmission, they can cause network signaling congestion due to frequent RRC (Radio Resource Control) transmission. In addition, the RRC transmission can cause extra delay and consequently can influence the achievable real-time performance of some small packets transmission. Therefore, 5G needs a proper mechanism to support transmitting this kind of packets [6].

Scheduling according to packet type (using, e.g., their priority, time-tag or collision probability) can be one of the proposed solutions. Managing and sharing processing resources between different queues in such a way that each queue behaves as if it has its own resources should be part of this solution (virtualization). It could reduce congestion probability and delay if

scheduling does not cause extra overhead to packets. So, some new scheduling without too much signaling overhead should also be studied, like Periodic scheduling of Keep-Alive packets [6].

The proposed solution can be related to the project due to its ability to reduce power consumption: using proper scheduling can accelerate packet transmission, prevent congestion and retransmission, because in the proposed solution, each kind of small packet requires specific resources, therefore this solution causes significant reduction in the amount of signaling and leads to faster message delivery.

2.3.2 Software-Defined-Networking

Software Defined Networking (SDN) is synonymous of a programmable network that has a separate centralized logically abstracted control plane and a flow-based data/forwarding plane. This simplifies the network and makes it more flexible and efficient. Besides, SDN provides Open APIs between the applications and the control plane, and between the control and data planes. Open APIs define functions to support, for example, mobility management of core networks [6].

In addition to separate radio access and radio core networks, SDN utilizes flexible combination and reload of functional building blocks. It means that software-defined content delivery supports distributing those functional building blocks across the network nodes based on needs [37].

Currently, the data/forwarding plane and control planes are mixed together, which increases processing requirements and provide less control, and consequently affects the error correction and re-transmission process, also regarding what is mentioned above and regarding increasing heterogeneous access demands to network, the 5G core network should leverage SDN to increase flexibility and scalability [33].

As OpenFlow-based SDN is the first standard for the SDN body and Wireless and the mobile working group of the Open Networking Foundation is currently working on SDN-based mobile packet core network, utilizing OpenFlow concept and standards in the wireless context could be a solution for this purpose. SDN and control/data plane separation are two of the most important concepts in the virtualization context. At first sight, it may seem that this subject is only weakly related to the project but, in order to leverage WAV, SDN could be a good solution because of its inherent features in terms of virtualization.

2.3.3 Virtualized mobile core network

Generally, the core network should have high speed and reliable capacity to be able to manage and deliver service to an increasing number of heterogeneous access technologies [38]. On the other hand, a mobile core network can be sliced into several overlay networks so that each slice serves different types of users.

In mobile core network, these characteristics are more important, so to increase capacity of mobile core network, using virtualized functions separated from hardware is a promising solution. Virtualized mobile core network manages functions and other resources to be more flexible and intelligent. Also, a virtualized platform can provide open APIs to management functions utilizing shared resources [6]. If we can perform NFV in order to manage mobile core network slices, it could be a good approach to implement a virtualized mobile core network. It is worth to mention that the core network is one of the most power consuming parts over the whole network and also that enough flexibility for 5G systems is needed so that every new access technology can be connected to the 5G core network without any change, thus virtualization could be useful to meet the goals of our project.

2.3.4 Virtualized C-RAN

Cloud RAN is an extended Radio Access Network (RAN) that China mobile introduced. On the other hand, the fixed mobile network components can be combined with NFV in order to centralize software applications on cloud-based hardware. C-RAN, in combination with SDN and NFV, has several advantages and the most important ones are:

- Extension of cloud service offerings from mobile core to RAN area.

- Flexibility with respect to integration of decentralized core functions in C-RAN processing units.

- Increased flexibility in end-to-end (E2E) services.

- Flexible integration and adaptation of Operation, Accounting and Maintenance (OAM) and Self-Organizing Networks (SON) functionalities. [37]

Due to C-RAN, a virtualized base station with several baseband units can be implemented on the same server as a virtual machine and supports different RATs. Today, a RAN consumes almost 70 percent of the total power of the network, therefore some power saving features are

essential parts of the design of energy-efficient networks such as using more energy efficient cooling by centralization of the baseband processing and the C-RAN network architecture [39].

A virtualized end-to-end solution from the core network to the RAN can enable the 5G goals of spectral and energy efficiency. Currently several operators have been developing and deploying green access infrastructure such as cloud/collaborative/clean radio access network (C-RAN) [39], [40].

Although enabling C-RAN as a set of virtualized functions seems essential, virtualization implementation should overcome critical real-time processing requirements, including load balancing and latency limits [6]. Improving energy efficiency by deploying C-RAN and especially virtualized C-RAN, and possibly increasing green energy consumption, achievable by using renewable energy for base stations, is related to the project.

2.3.5 New waveforms

OFDM with some enhancements such as numerology is sufficient for up to 20-30 GHz frequency range, but for frequency above 30GHz, it should be re-considered. Therefore, for the 5G network, there is a need to enhance OFDM or introduce alternative to be used as a new multiple access technique. In order to do that, each method must be explored in terms of performance improvement and for specific requirements or scenarios. These methods have some potential benefits to the 5G network that should be considered. New waveform can enable steeper spectrum roll-off and bands with difficult constraints. Besides, it is able to reduce power consumption and help to have energy efficiency and cheaper devices, especially for machine to machine (M2M) technology. New waveforms can also enable low latency framework and high transmission range in higher frequency bands. Then it can be combined with massive MIMO in higher frequency bands that leads to deliver higher spectral efficiency [6].

Some methods that should be considered as solutions for new waveform are Generalized Frequency Division Multiplexing (GFDM), Filter-Bank Multi-Carrier (FBMC), Faster Than Nyquist (FTN), Wave Amplitude Modulation (WAM), Sparse Code Multiple Access (SCMA), Filtered-OFDM (F-OFDM), Universal Filter Multi-Carrier (UFMC). In addition, enhancements of OFDM (modified Cyclic Prefix) and single carrier (SC) FDM should be taken into account as well. From these mentioned methods, UF-OFDM improve spectral properties and robustness to time and

frequency but for mm-Waves there is a need to have different waveforms. SCMA can provide new properties to flexibly support diversity of use case. Therefore, SCMA can be a good choice as a new multiple access technique for 5G.

2.4 SCMA Implementation

To provide massive connectivity, better quality of service, higher throughput, lower latency, or lower control signaling overhead, Sparse Code Multiple Access (SCMA) was proposed as a new frequency domain non-orthogonal multiple-access technique, in order to improve spectral efficiency of wireless radio access [42]. SCMA re-uses non-orthogonal code domain resources. This technique can provide massive connectivity, thus it is able to serve a large number of users and improve spectral efficiency of radio access [44].

In SCMA, each incoming data stream is represented as a codeword. This codeword comes from different multi-dimensional codebooks and demonstrate a spread transmission layer [41] [42]. Each codeword represent a transmission layer. In comparison with QAM symbols in LDS, SCMA improves performance because of multidimensional constellation.

In addition, direct mapping of incoming streams to multidimensional codewords of SCMA codebook sets is performed by combining two fundamental procedures, QAM symbol mapping and spreading [41] [43]. Therefore, in comparison with LDS, SCMA can offer a better coding in terms of multi-dimensional constellations than simple repetition of LDS codes and consequently, SCMA is able to improve spectral efficiency [41]. In addition, as signals transmitting each codeword consume different power, MPA can cancel inter-layer interferences more efficiently.

In [41], a design approach of SCMA codebooks is proposed. Part of its complexity stems from using different codebooks for multiple multiplexed layers. The proposed approach starts by a multidimensional constellation and continues by rotating the base constellation to find a proper product distance. To simplify design of multi-dimensional codewords, authors in [41] define a mother constellation and the operators separately. They also define Design Metrics, Rotated Constellations, Rotated Lattice Constellations, Shuffling Multi-dimensional Constellation Imaginary Axes and Rotation to Minimize the Number of Projection Points. As a result, SCMA leverages features of LDS, like overloading, moderate complexity of detection by using a MPA and interference whitening, while improving the link performance. Thus, it can offer better

performance in both uplink and downlink multiple access scenarios, which is needed for next generation wireless networks [41].

The first step to implement SCMA can be designing its codebooks. In order to design the SCMA codebooks, one way is leveraging a systematic multi-stage method that is proposed in [6]. The next step is defining uplink and downlink chains. In comparison with LTE SC-FDMA transmission system for uplink, it would be possible to leverage the LTE SC-FDMA transmission system while replacing QAM modulation and DFT modules with SCMA encoder in the transmitter section and replacing the receiver and modulation de-mapper with a SCMA decoder in the receiver section. In addition, a Message Passing Algorithm (MPA) detector should be implemented as a complement for SCMA decoder in the SCMA receiver.

Another possibility is implementing the SCMA-based uplink grant-free multiple access that is proposed in [11]. In SCMA, coded bits are mapped to codewords in the complex domain. Then, spreading codewords are transmitted to the channel. In the receiver, multi-user detection and channel decoding are applied. Sparsity makes SCMA a relatively low complexity algorithm. In this algorithm, the number of codewords can be much larger than the number of resource units which open possibilities to handle more users.

Considering SCMA apparent potential, it can be a proper solution as a new multiple access technique for 5G networks. Therefore, in order to enable using it, there is a need to analyze SCMA specifications at first, then verifying SCMA performance.

2.4.1 SCMA a Solution Proposed for Improved 5G Transmission

Wireless networks are evolving toward a fifth generation that aims at supporting massive connectivity of different devices and diverse applications at the same time, beyond the capacity offered by LTE. In that enhanced environment, the cost of signaling overhead and latency should be taken into account for further investigation [11].

In current wireless networks, multicarrier CDMA is used to spread modulated QAM symbols using orthogonal FDMA (OFDMA). In addition, an approach with CDMA called Low-density signature (LDS) performs an almost optimal MPA receiver with practical complexity, because LDS spreads large signatures with a small number of non-zero elements in it. The

performance offered by MPA is claimed to be good even for overloaded systems [41].

The properties of SCMA proposed in [43] include encoding binary data to multidimensional complex codewords, generating multiple codebooks for each layer or user, detecting multiplexed codewords by the MPA multi-user detection technique and providing a system overloading capability.

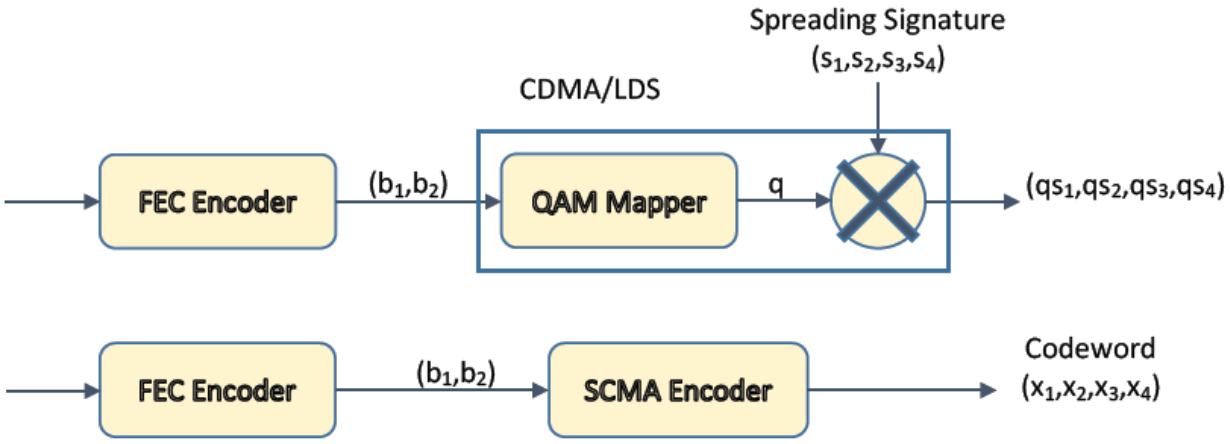

Figure 2.1: Merged QAM mapper and Spreading in SCMA waveform

CDMA spreads the data symbols over orthogonal code sequences and in the receiver, the original data symbols are extracted using an orthogonality feature. The important factor in a CDMA system is the sequence design that influences the performance and the complexity in the receiver. A CDMA modulator in the encoder is responsible to map a number of coded bits or QAM symbols to a sequence of complex symbols. Therefore, by merging the QAM mapper and CDMA spreader, a number of bits can be directly mapped to a complex vector, which is a coding procedure, and it is the exact way SCMA works (Figure 2.1).

Some of SCMA properties are mentioned in [43]. These properties include directly mapping binary data to multidimensional complex codewords, multiple access by using multiple codebooks for different users, using MPA multi-user detection technique because of codebooks sparsity feature and having multiplexed layers in larger numbers than the spreading factor. Since optimal SCMA codebook design is very complicated for multiple users with different codebooks, authors in [43] propose a sub-optimal systematic design. In the SCMA encoder with optimal codebooks, binary bits are mapped to multidimensional complex sparse vectors with nonzero

indexes called codewords, otherwise codewords consist of zero values. Then, SCMA multiplexer multiplexes codewords over orthogonal physical resources. The SCMA code can be presented with factor graph matrix as well. In the receiver, a MAP algorithm detects the layer codewords with the channel knowledge and receiving data.

In comparison with LDS, SCMA uses codebooks to carry the information over multidimensional constellation points by replacing modulation and sequences while LDS using spread QAM symbols to carry information. This influences SCMA performance by using multidimensional constellation points. Simulation results indicate better performance for SCMA in terms of BLER compare to LDS [43].

Implementing SCMA introduce two challenging problems associated with the need to develop a multidimensional lattice constellation and effective SCMA codebooks. In [43], the authors proposed a systematic approach to resolve the second problem. The proposed system is decomposed in a number of steps associated with the development of mechanisms to implement a SCMA Encoder, SCMA Multiplexing, Factor Graph Representation, a SCMA Receiver and means to use SCMA versus LDS separately. The optimal criteria to design SCMA code is not determined, so in [43] a multistage optimization approach is proposed. The optimization is performed in several stages associated with: the Mapping Matrix, the Constellation Points, the Mother multidimensional constellation and the Constellation function operators. In this approach, the MPA detects and separates interfering symbols at a source node based on their power levels, so for this purpose, a mother constellation must provide different average power levels for distinct constellation dimensions.

One of the proposed multiple access techniques in a downlink wireless network is multi-user MIMO. MU_MIMO shares time-frequency and power resources between multiple users that improves in spectral efficiency [44] [45]. An important factor influencing performance gain with MU-MIMO concerns cross-layer interference, which sets a requirement for estimating channel state information (CSI). In [42] a multi-user SCMA (MU-SCMA) is proposed for downlink wireless access to improve the network throughput. In addition, this design includes power sharing, rate adjustment and scheduling algorithms to have a better downlink throughput for a network that is heavily loaded. Also, SCMA advantages are evaluated in terms of its performance for lightly loaded networks. Generally, multi-user MIMO (MU-MIMO) is utilized to share power resources

and frequency in order to increase throughput through user multiplexing [12]. MIMO layers are separated orthogonally in the space domain and each layer is assigned to a user. Therefore, users at the receiver can easily find their layer with no cross-layer interference that makes detection at user nodes much simpler. Besides, MU-MIMO provide a high throughput gain but it is a closed loop system and has some disadvantages like high overhead to feedback CSI about users, thus MU-MIMO performance gain depends on CSI procedures. In order to resolve the MU-MIMO limitation, using an open-loop multiplexing can be a proper solution and a non-orthogonal multiple access like SCMA is an open-loop scheme [42] [46]. In SCMA, as this system has an open loop multiple-access scheme, CSI feedback does not cause any problem because there is no need for CSI knowledge of paired users to allocate code –domain layers.

In [42], instead of MU-MIMO, a multiuser SCMA (MU-SCMA) scheme is proposed. MU-SCMA is more robust against channel variation than MU-MIMO and it does not need to have information about channel quality as well. As explained in [42], the main concerns in multiusers system for lightly loaded networks are interference and robustness. In LTE networks, which use the OFDMA system, when the traffic is low, some resource blocks are muted to better manage resource utilization. In general, at each resource block, the interference level goes up if the neighbor cells are occupied and it goes down if the neighbor cells are empty. Using SCMA instead of OFDMA makes the interference white that improves link robustness. In an SCMA system, link quality depends on the number of layers, codebook sizes, coding rate and power level of multiplexing layers.

To evaluate SCMA performance in a downlink wireless network, two scenarios are used in [42]. The first scenario is a fully loaded network with high throughput demand and the second one is a lightly loaded network with variations in each scheduling interval. In order to evaluate the throughput improvement of MU-SCMA, a heavily loaded network is considered. However, to evaluate interference averaging, a lightly loaded network is considered. Since SCMA is not a linear modulation, in [42] the linear sparse sequence modeling is used to develop MU-SCMA related algorithms, and consequently, an equivalent model for MIMO in the linear sparse sequence system is developed. In addition, several algorithms are needed to enable MU-SCMA. The “User Pairing to Maximize Weigthed Sum-Rate” algorithm is used to decrease the pairing complexity. In the “Rate Adjustment and Detection Strategy” algorithm, two strategies that are considered are detection of high quality users and detection of low quality users from which an optimum operating

point can be found. Other algorithms are used to optimize OFDMA and SCMA power sharing factors.

Simulation results characterizing SCMA and MU-SCMA are presented in [42]. One of the main advantages of open loop MU-SCMA in comparison with MU-MIMO is the robustness when the system fails because of high mobility speed of terminals in a network. In comparison with OFDMA, SCMA in a heavily loaded network has 5 percent higher throughput and 8 percent higher coverage gain. The main reason for those findings is the inherent feature of SCMA with its multidimensional codebooks and its flexibility in link adaptation. Besides, when MU-SCMA is used instead of OFDMA, the throughput and coverage gain increase about 28% and 36% respectively. The simulation results of lightly loaded network, when 50% of resources are utilized, show the better performance for SCMA compared to OFDMA, and the throughput and coverage gain increase by about 56% and 26% respectively. These results show that SCMA can improve robustness of lightly loaded networks. The MU-SCMA used in [42] is for single-TP and SIMO channel that can be extended for multi-TP and MIMO systems. Consequently and considering all the results, MU-SCMA is a good choice for future wireless networks.

In [44], a combined technique is mentioned that is the extension of spatial multiplexing. This technique is code-space-multiplexing (CSM) that is a combination of SCMA with MIMO technique. The base station in massive-CSM system (massive MIMO MU-SCMA), has massive transmit antennas and transmits SCMA codewords for each user with single antenna. SIMO SCMA is another CSM system where base station and the user equipment are equipped with single antenna and multiple antennas respectively. The CSM system improves capacity and spectral efficiency in both SIMO and MIMO version. In massive CSM system, a new codeword is created by overlapping codewords of each layer together and mapped to massive transmit antenna using a linear matrix. An asymptotic equivalent of signal to interference plus noise ratio (SINR) is gained when the number of transmit antennas approaches infinity by using random matrix theory. This equivalent is used to analyze the capacity for massive CSM system. Simulation results show that massive CSM has higher sum rate performance in comparison with massive MIMO MU-OFDMA.

The SCMA system in [44] considered for simulation is a synchronous downlink system with K single antenna users. In this system, a base station has M transmit antennas with M much greater that K. The number of transmit layers and orthogonal resource blocks are assumed to be J

and D respectively. Considering these assumptions, massive connectivity is possible because large overloading factor can be used. In CSM system, the authors assumed K=J, thus assigning different layers to different users allow serving more users. For downlink, a massive CSM system is considered. At first, it is an asymptotic equivalent in term of SINR, which is derived to improve the asymptotic sum rate of massive CSM. In order to do simulation, two cases are considered: The first case is Single user CSM that is an SCMA with single user. This case assumes one user with J layers. The second case is Equal Average Power for massive CSM that distributes power among SCMA codewords. In the first case, different layer interference happens in CSM and consequently causes a lower sum rate performance in the SCMA without multi-user diversity. Simulation results in this paper indicate that when the number of transmit antenna goes up, the sum rate is increased as well. In comparison with MIMO MU-OFDMA, massive CSM sum rate is higher because of the sparsity of codewords. In addition, it appears from simulation results that the sum rate is getting higher when the number of antennas is increased, and results of analytic expression approach theoretical values for the SINR.

Uplink transmission in LTE is performed by serving base station with a request grant procedure in this case a UE sends a scheduling request (SR) to the network periodically in order to obtain dedicated uplink resources. It usually takes about 7ms or even more between the SR and uplink data transmission [47]. Such delay will not be tolerated for future applications, especially for delay sensitive traffic, so one approach to reduce the latency and signaling overhead in uplink is using contention based data transmission [44].

In [11], some important properties of an uplink contention based SCMA system are mentioned. The SCMA contention based transmission mode has just a few or even one predefined time-frequency regions. The dedicated SCMA codebook characterizes each user or each layer, which represents a user. In fact the SCMA codebook is a user's contention region. The SCMA codewords, collected for incoming streams from the SCMA codebook are sparse to attenuate the complexity of MPA used for multi-user detection [48] [49]. Another advantage of sparsity of SCMA codewords is the overloading feature of SCMA. If the number of multiplexed layers is more than the length of the codewords, the system can be overloaded but since the SCMA codewords are sparse, massive connectivity with less detection complexity can be provided [11].

In [11], an uplink contention based SCMA is proposed and means to design the PHY and

MAC layer are described. A contention transmission unit (CTU) is the basic resource for contention transmission, while if two or more users use the same CTU, it causes user collision that should be solved by random back-off mechanisms. The proposed solution for uplink contention based SCMA has two parts. The first part is a multiple access mechanism based on multiple UEs contention for the same resource (CTU). The second part is non-adaptive transmission that has predefined sets of coding and modulation levels.

Comparing SCMA and OFDMA, both deliver similar QoS when they work in different traffic load, but SCMA is able to support a larger number of active users when compared to OFDMA. As explained before, the better performance of SCMA is due to the sparsity of the SCMA codewords and better multiuser detection with MPA.

2.5 Generic Tools and Methodology for Hardware Accelerators Development

This project aims at performing design space exploration for possible hardware implementations of several functionalities required for 5G and HETNET-WAVES. This could accelerate development and commercialization of new telecommunication products. Approaches that allow reducing design cost are of general interest. Design reuse and raising the abstraction level are two of the most important concepts leading to design acceleration. In order to achieve these goals, some generic tools and methodology were utilized. Implementation typically starts with MATLAB simulations and continues with Software implementation using the C programming language. Leveraging C-based codes as a base of HLS implementation is a logical next step to design possible hardware accelerators, at least in a research environment where only the best algorithmic and architectural solutions are considered for actual hardware implementation. Thus, in a research environment such as this project, cutting development time is critical, as many solutions are considered, but never make it on an actual deployed implementation.

2.5.1 Programming tool

Programming tools for engineering purposes often face two different levels of need. The first one is when there is a need to solve a problem and computational efficiency is not critical. In this case, usually the execution time may be of secondary importance, while accuracy and ease of

development are the main concerns. Therefore, at this level, using engineering environments like MATLAB can be a good choice. However, for the second level, program execution time is a concern. This is particularly true when a program is a part of a real time system. Thus, for programming at this level, an efficient language should be used. Another thing is programming at second level should be based on a recognized standard, and programming languages like C or C++ are proper choices. C is a high level structured programming language that is efficient and powerful, particularly in the embedded system areas. On the other hand, C is also a middle level language that has both the advantages of low level and high level languages. Another advantage of the C language is that C can be compiled on different platforms such as Xilinx Vivado HLS.

2.5.2 High Level Synthesis

Xilinx Vivado High Level Synthesis (Vivado HLS) is a tool that allows synthesizing digital hardware directly from C-based high level descriptions. This tool helps to avoid creating hardware designs manually with hardware-description languages such as VHDL or Verilog. Leveraging HLS, the design functionality and its hardware implementation are separate, because C-based descriptions do not impose fixed hardware architectures, like those specified by Register Transfer Level (RTL) specifications, and therefore it provides greater flexibility and time savings particularly, for design space exploration [50]. In addition, HLS implementations offer a high productivity and can offer remarkably good results.

There are four different levels of abstraction for current FPGA design: structural, RTL, behavioral, and high-level (Figure 2.2). Structural HDLs (like VHDL or Verilog) are expressed in terms of specific interconnected components. Register Transfer Level (RTL) specifications hide technological details and represent designs as operations occurring between registers. A higher level of abstraction is behavioral HDL that is an algorithmic description of the circuit with which designs could be done faster. Finally, HLS is a design method that is not based on an HDL, but that rather express a design with the C based abstraction [50].

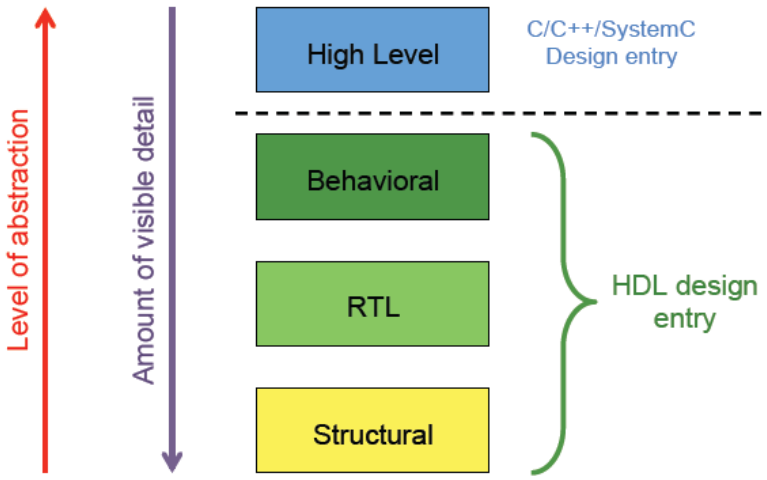

Figure 2.2: Levels of abstraction for current FPGA design [55]

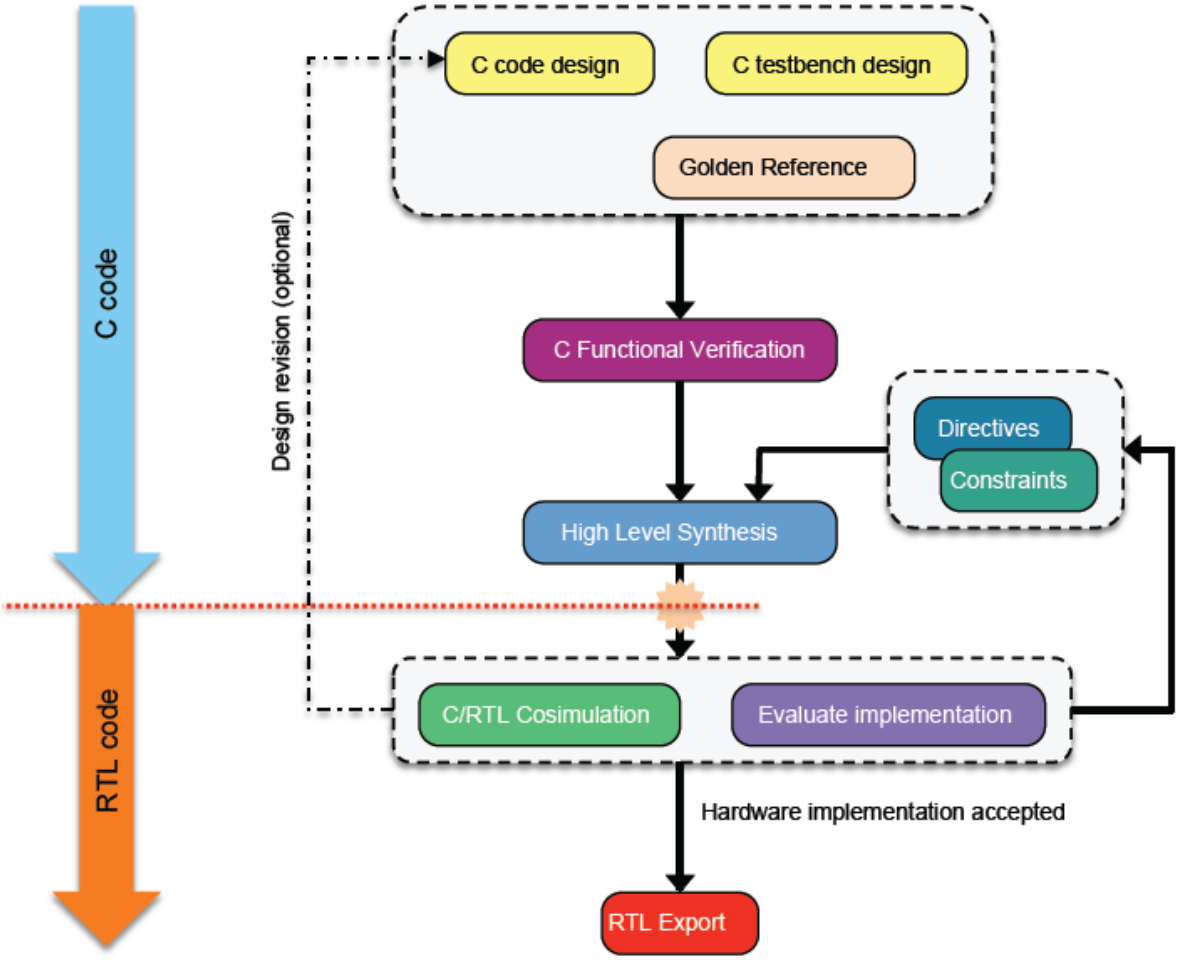

Figure 2.3: An overview of the Vivado HLS design method [55]

What Vivado HLS does is transforming a C based design to an RTL implementation, which is then synthesizable and can be implemented on Xilinx FPGA devices. The Vivado HLS design flow consists of the primary processes like designing the C code and C testbench and the related outputs, the execution of HLS algorithms and the production of RTL code and elements for verification (see Figure 2.3) [51].

2.5.2.1 Inputs to the HLS Process and Functional Verification

A C-based function is the input to the HLS process. To verify this function, a C based testbench is needed. Verifying the functionality of C based code should be done before the synthesis process begins. This verification can be done by performing a C code simulation. C simulation application is available in Vivdo HLS tool and helps to verify the C design, characterize the output results and modify the C code if it is necessary.

2.5.2.2 High-Level Synthesis

Analyzing and processing C based code is performed according to user-supplied directives and constraints. The inputs to high level synthesis are C/C++ or SystemC files, C testbench files, constraints and directives. The constraints and directives inputs influence synthesis process along together. Constraints include clock period and details of the target device. However, directives determine the style of implementation like pipelining and parallelism. In fact, the outputs of high level synthesis can be SystemC models, VHDL or Verilog files and Packaged IP for Vivado or System Generator. After the HLS process, design files in the desired RTL language are produced. In addition, various other log and report files, testbenches, scripts, etc. are created (Figure 2.4).

2.5.2.3 C/RTL Co-simulation and Evaluation of Implementation

C/RTL co-simulation in Vivado HLS helps to verify the equivalent RTL model produced against the C based code. With this process, the C based testbench is used to provide inputs to the RTL version and the outputs are checked against expected values, without a need to generate a new RTL testbench. The RTL testbench is created automatically by Vivado HLS and it manages data passing between C-based testbench and RTL module. The RTL output can be evaluated in terms of the amount of resources required to implement it in hardware, the processing latency of the design and the maximum clock frequency at which it can operate [51].

2.5.2.4 RTL Export

After RTL validation, there is a need to integrate the design into a larger system that is synthesizable from RTL files (VHDL or Verilog code) created automatically by the HLS process. Also, Vivado HLS is able to produce IP packaging and pass the design to other Xilinx tools like IP integrator or system generator.

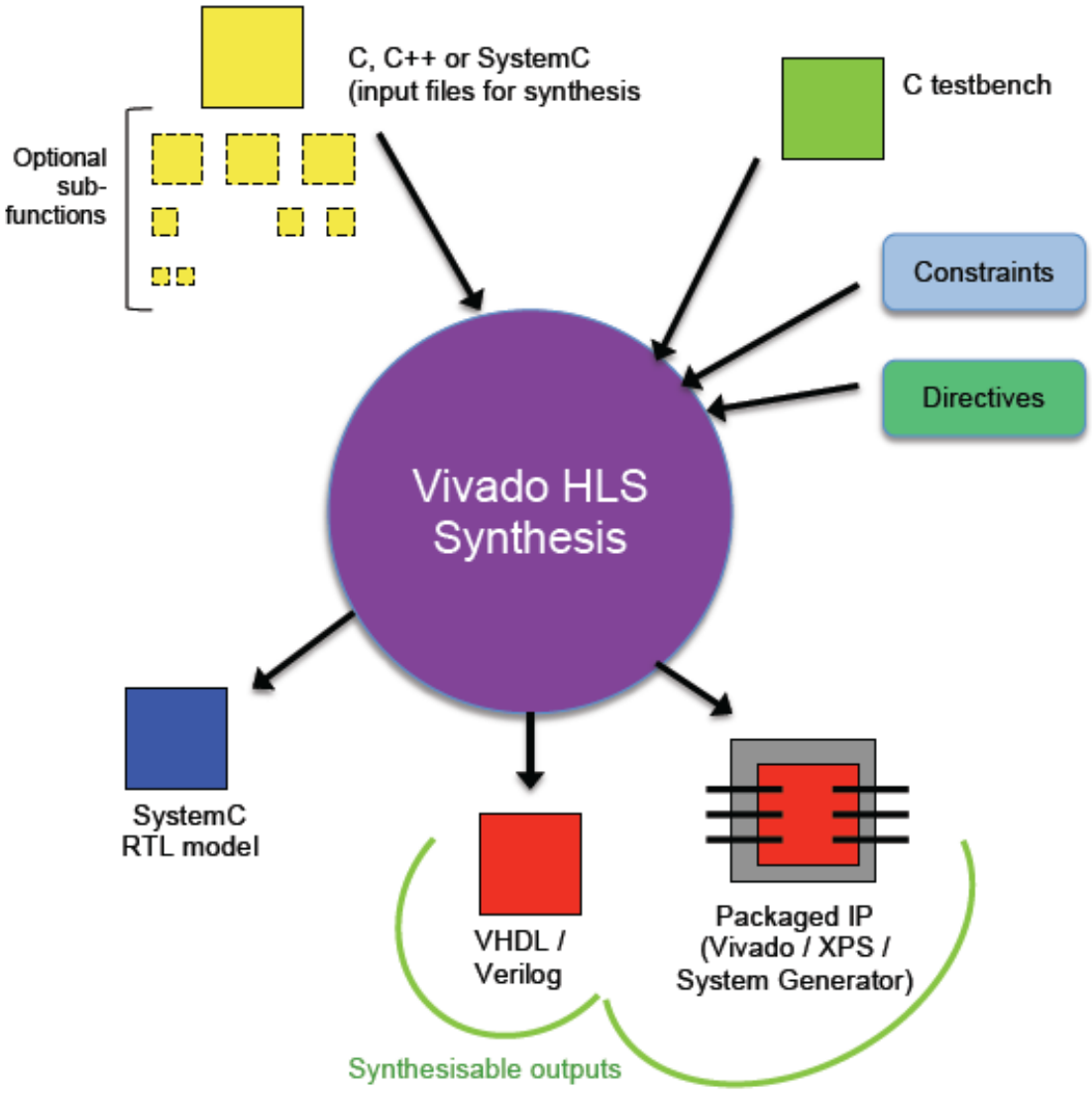

Figure 2.4: An overview of the Vivado HLS synthesis process [55]

2.5.2.5 Implementation Metrics

The most important metrics characterizing resulting designs are resources and area, throughput, clock frequency, latency, power consumption and I/O requirements. The priority of

each metric depends on the application requirements. However, application needs constraints that are chosen considering one or some of above factors [50] [51].

- Resources or Area: the hardware cost of building a circuit for an application on the FPGA is defining the main part of the design cost. This cost has a direct relation with how much resources or area the deign consumes on the FPGA. Therefore, the consumed area should be preliminary measured. In addition, somehow minimizing the cost and consequently the area is usually important. Vivado HLS tries to minimize the area by default. Notably, it tries doing so by time sharing hardware modules. This influences latency and throughput.

- Clock Frequency: the maximum clock frequency is related to physical characteristics of the target device and critical path of the RTL design produced by the synthesis process. The clock period is a target constraint that should be defined by the user.

- Latency: Latency can have different definitions for different purposes. In Vivado HLS, latency is the number of clock cycles between applying inputs and producing related outputs. The maximum acceptable latency can be defined as a constraint to the design.

- Throughput: The data rate that can be passed through the system is the throughput.

2.5.2.6 Directives

One of the techniques to optimize the design in Vivado HLS is using different directives. Some of these directives are ARRAY_MAP, ARRAY_PARTITION, DATAFLOW, LATENCY, LOOP_MERGE, FUNCTION_INSTANTIATE, PIPELINE AND UNROLL. The way directives are used depends on design requirements [51]. In order to optimize a design in Vivado HLS, there is a typical way to use constraints and directives. The optimization should start with clock period constraints and followed by directives. For instance, pipelining the tasks improves throughput and loop transition removal improves latency.

2.6 Summary of Literature Review

Reference [6] provides a broad overview of 5G challenges and requirements. It provides a context and a set of useful guidance for the research that will be conducted in this thesis. This chapter reviewed numerous 5G research topics and wireless network virtualization that can be useful in 5G networks. Five subjects of interest were identified among the 14 different challenges

that are discussed in the NGMN white paper. One of these topics relates to proposing new multiple access technique like SCMA. This research explores means of implementing SCMA encoders with HLS tools and design methods that could then be applied to other functional modules needed in 5G processing chains.

CHAPTER 3 SCMA ENCODING IMPLEMENTATION EXPERIMENTS

In order to improve wireless access, new modulation techniques such as SCMA are considered. In order to assess their suitability, possible hardware platform architectures are considered. Final solutions must meet the technological constraints imposed by clock speed, transistor density and die size to support ever increasing data throughput requirements and complexity of advanced signal processing techniques. In typical wireless system implementations, higher-level functionalities are implemented in software or firmware executed on general purpose central processing units (CPUs), while the PHY and MAC layers functionalities are generally executed on a mixture of DSP cores and dedicated baseband accelerators. However, virtual base stations can also be implemented by integrating several systems on chip (SoC) with a high-speed interconnect fabric.

Fig. 3.1 shows the functional diagram of a 4G system where some turbo code implements channel coding and OFDMA is used as a multiple access technique. Regardless of 4G-LTE, 5G system require new technologies that can improve throughput, channel capacity and reliability. A proposed simple system for 5G is shown in Fig. 3.2.

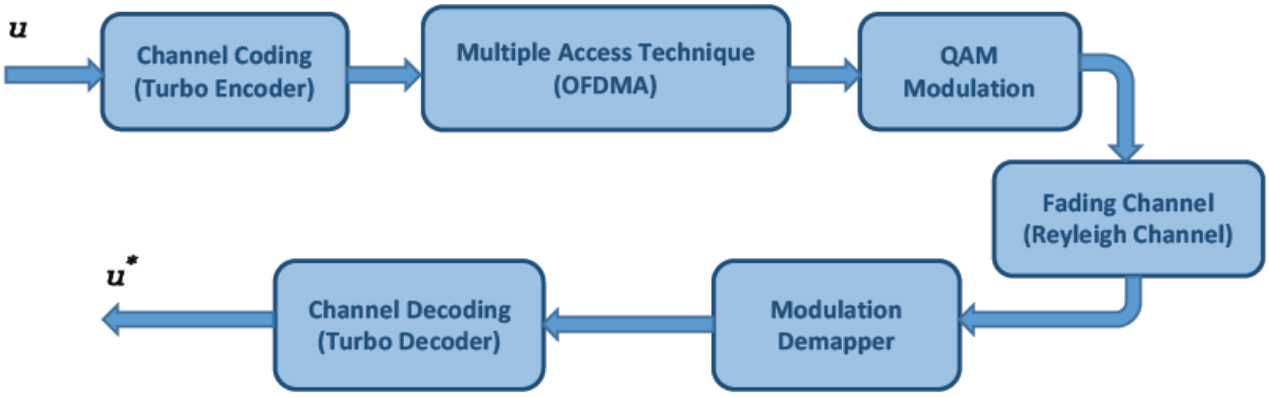

Figure 3.1: An overview of 4G system technology

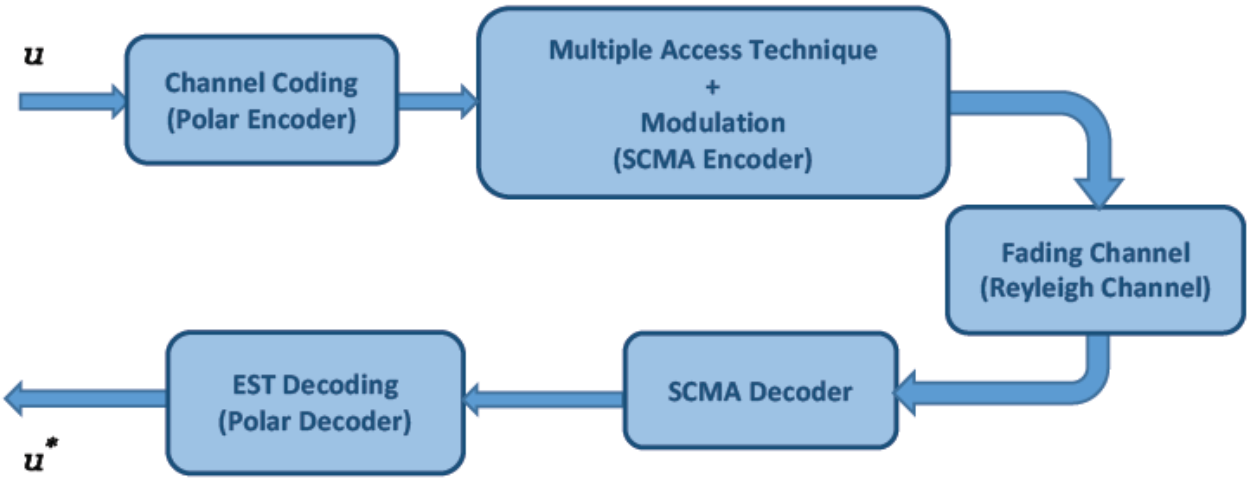

Figure 3.2: Possible technologies for 5G system

As shown in Figure 3.2, a promising solution for 5G is to use new multiple access techniques such as Sparse Code Multiple Access (SCMA) as means to provide multiple accesses to the channel. SCMA is a recently proposed multiple access technique called that is a frequency domain non-orthogonal multiple access technique proposed in order to improve spectral efficiency of wireless radio access [42]. In SCMA, each incoming data stream is represented as a codeword. The codewords come from different multi-dimensional codebooks implementing a spread transmission layer [41] [42]. In addition, direct mapping of incoming streams to multidimensional codewords of SCMA codebook sets is performed by combining two fundamental procedures, QAM symbol mapping and spreading [41] [43]. The properties of SCMA proposed in [43] include encoding binary data to multidimensional complex codewords, generating multiple codebooks for each layer or user, detecting multiplexed codewords by the MPA multi-user detection technique and providing a system overloading capability.

In the set of algorithms considered for the new multiple access techniques including encoding and decoding algorithms, SCMA encoding is the simplest and was used as means to experiment and shows the benefits of generic implementation methods discussed this chapter. One of the main concerns of using a new multiple access technique like SCMA is to provide lower latency and consequently higher throughput. SCMA encoding was chosen because of its simplicity and lower processing complexity. Therefore, using SCMA encoding can reduce energy required for data transmission while it provides high throughput.

This section is dedicated to explore various means to improve SCMA encoder

implementations in software and in hardware. The goal was achieving a throughput of more than 1Gbps. More than 1Gbps throughput was chosen considering throughput of 4G LTE network that is available nowadays. The rest of this chapter is thus dedicated to explore complexity and means of implementing SCMA encoders in software and in hardware.

In a 5G transmission chain, information bits encoded by a channel coder such as turbo encoder or polar encoder arrive to an SCMA encoder. This encoder maps the encoded bits to I and Q complex values using codewords that are then transmitted through the channel. Codewords are selected from multidimensional codebooks. In our implementation, there are six data layers/users and four physical resources. Therefore, each codeword constellation point is transmitted over one of four existing physical resources (PREs). Since there is no data dependency between the six data layers/users defined in the SCMA encoding algorithm, data layers/users incoming bits can be processed in parallel, and we leverage this feature in the implementations reported in this chapter. Thus, each physical resource (PRE) transmits the summation of six constellation points from six different data layers as illustrated in Figure 3.4.

3.1 SCMA Transmitter Chain

SCMA codebooks constitute users contention regions that characterize each user or each data layer. The SCMA codewords, collected for incoming streams from the SCMA codebook are sparse to mitigate the complexity of the Message Passing Algorithm (MPA) used for multi-user detection [48] [49]. A simple SCMA transmitter chain, as illustrated in Figure 3.3, includes three modules: the Turbo Encoder, the SCMA Encoder and the PRE Mapper.

Designing the codebooks can be considered as the first step to implement SCMA, from which uplink and downlink chains can be defined. However, if finding optimal codebooks is a challenging open problem related to communication theory, good codebooks that can be used in practical implementations are readily available. Thus the issue of codebook design is not considered further in this thesis. An example of a codebook set including six data layers/users for four physical resources is used in the following to explain the concept [52].

$$Codebook\ 1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -0.1815 - 0.1318i & -0.6351 - 0.4615i & 0.6351 + 0.4615i & 0.1815 + 0.1318i \\ 0 & 0 & 0 & 0 \\ 0.7851 & -0.2243 & 0.2243 & -0.7851 \end{bmatrix}$$

$$Codebook\ 2 = \begin{bmatrix} 0.7851 & -0.2243 & 0.2243 & -0.7851 \\ 0 & 0 & 0 & 0 \\ -0.1815 - 0.1318i & -0.6351 - 0.4615i & 0.6351 + 0.4615i & 0.1815 + 0.1318i \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Codebook\ 3 = \begin{bmatrix} -0.6351 + 0.4615i & 0.1815 - 0.1318i & -0.1815 + 0.1318i & 0.6351 - 0.4615i \\ 0.1392 - 0.1759i & 0.4873 - 0.6156i & -0.4873 + 0.6156i & -0.1392 + 0.1759i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Codebook\ 4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.7851 & -0.2243 & 0.2243 & -0.7851 \\ -0.0055 - 0.2242i & -0.0193 - 0.7848i & 0.0193 + 0.7848i & 0.0055 + 0.2242i \end{bmatrix}$$

$$Codebook\ 5 = \begin{bmatrix} -0.0055 - 0.2242i & -0.0193 - 0.7848i & 0.0193 + 0.7848i & 0.0055 + 0.2242i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.6351 + 0.4615i & 0.1815 - 0.1318i & -0.1815 + 0.1318i & 0.6351 - 0.4615i \end{bmatrix}$$

$$Codebook\ 6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.7851 & -0.2243 & 0.2243 & -0.7851 \\ 0.1392 - 0.1759i & 0.4873 - 0.6156i & -0.4873 + 0.6156i & -0.1392 + 0.1759i \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3.1.1 SCMA Encoder

As explained before, in an SCMA Encoder, coded bits are mapped to codewords in the complex domain as amplitudes of I and Q signals. Then, spreaded codewords are transmitted through the channel. The coded bits that are encoded by turbo coding are injected at the SCMA encoder input. Each two incoming bits are mapped to a codeword that comes from different multidimensional codebooks. As we used a sample codebook set, for each data layer/users, a different codebook is used. Each codebook has four codewords that is a vector of four constellation points. Therefore instead of transmitting two bits, four constellation points are transmitted.

Each codeword has four constellation points transmitted over four respective physical

resources (PREs). In our implementation, there are six data layers/users and four physical resources. Also, Incoming bits for each data layer are independent and can be processed in parallel, so that each physical resource (PRE) transmits the summation of six constellation points extracted from codewords of six different data layers.

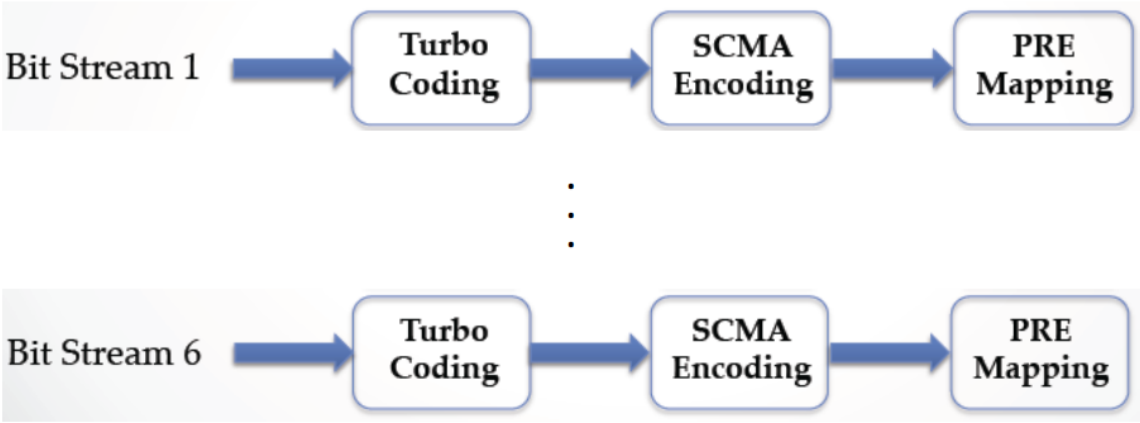

Figure 3.3: A Simple SCMA Transmitter Chain

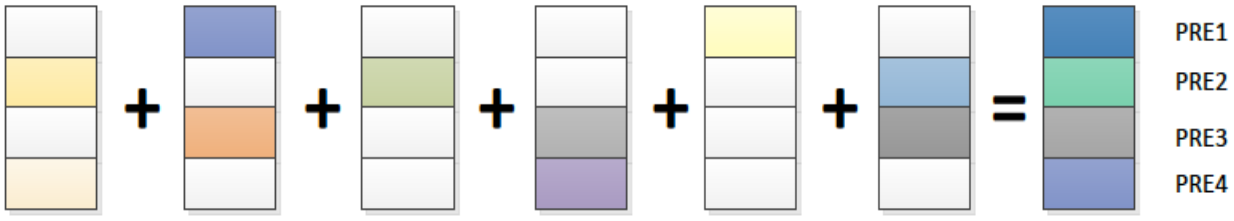

Figure 3.4: PRE mapping

3.1.2 PRE (Physical Resource) Mapping

Each codeword has four constellation points and there are four physical resources (PREs) as well, in this case each constellation point of a codeword would be transmitted over a PRE. Since in the implementation, there are six data layers/users and four physical resources, to transmit each element of a codeword from six users over a PRE, six constellation points should be added together and then transmitted. Also, incoming bits for each data layer are independent and processed in parallel, so what would be mapped to each physical resource (PRE) is the summation of six constellation points of six different data layers.

3.2 MATLAB Implementation

MATLAB was used for preliminary implementation of an SCMA encoding chain in conjunction with Turbo encoders available in the communication toolbox. Thus, the turbo encoder from the MATLAB communication toolbox can be leveraged in conjunction with the SCMA transmission chain. Input data for each user is different and independent and has at least 1024 bits. As our assumption in this implementation is to have six data layers/users and four physical resources, six different shapes and colors were used to display constellation points related to each data user in Figure 4.1 in Chapter 4. The implementation codes are reported in Table 3.1, Table 3.2 and Table 3.3.

Table 3.1: Turbo Encoding using MATLAB Communication Toolbox

Turbo Encoding using MATLAB Communication Toolbox
1: *EbNo = -6;* 2: *frmLen = 1024;* 3: *rng default* 4: *noiseVar = 10^(-EbNo/10);* 5: *intrlvrIndices = randperm(frmLen);* 6: *hTEnc_1 = comm.TurboEncoder('TrellisStructure',poly2trellis(4, ... [13 15 17],13), 'InterleaverIndices', intrlvrIndices);* 7: *hMod = comm.BPSKModulator;* 8: *hChan = comm.AWGNChannel('EbNo',EbNo);* 9: *data = randi([0 1],frmLen,1);* 10: *encodedData_1 = step(hTEnc_1,data);* 11: *modSignal_1 = step(hMod,encodedData_1);* 12: *receivedSignal_1 = step(hChan,modSignal_1);*

By contrast, what will be transmitted over each physical resource (PRE) is complex amplitudes of I and Q orthogonal carrier signals. As six constellation points should be transmitted through a PRE, a total of six I and Q signals will be transmitted thorough a PRE. Therefore, the

summation of six constellation points of six codewords of different data layers/users are transmitted over PREs (See Figure 4.2, Chapter 4).

Table 3.2: Codeword Mapping for one user

Codeword Mapping for one user
1: **for** *j=1:2:(s1-1)* **do** 2: *CW_U1(:,j)=SCMA_EN(En_in1, CB_1);* 3: **end**

Table 3.3: PRE (Physical Resource) Mapping

PRE (Physical Resource) Mapping
1: **for** *i=1:(s1-1)* **do** 2: $PRE(:,i) = \sum_{n=1}^{6} CW\_Un\ (:, i)$; 3: **end**

3.3 SCMA Encoder Software Implementation and Results

In order to assess the complexity of SCMA encoding and multiplexing for PRE mapping, a software implementation was developed with the C programming language. This implementation assumes six data layers/users and four physical resources like the MATLAB implementation. A compatible sample codebook set was used [52].

As a basic SCMA encoder encodes two bits of each data layer/user per iteration, the first software implementation followed this rule. In this implementation, as soon as the first two bits arrive, the encoding process starts. SCMA encoding pseudo code for this implementation is reported in Table 3.4. The processing time of SCMA encoding and PRE mapping was measured (See Table 4.1, Chapter 4).

To improve the processing time, there is an opportunity to encode more than two data user bits per iteration.

Table 3.4: SCMA Encoding pseudo code for the first implementation

SCMA Encoding pseudo code for first implementation
1: /* SCMA Encoding Function*/ 2: **void** *SCMA_En (char in_bit [2], double _Complex in_cb[4][4], double _Complex out_cw[4])* 3: { 4: **if** *(in_bit == 00) {* 5: *mapping codeword vector1 }* 6: **else if** *(in_bit == 01) {* 7: *mapping codeword vector2 }* 8: **else if** *(in_bit == 10) {* 9: *mapping codeword vector3 }* 10: **else if** *(in_bit == 11) {* 11: *mapping codeword vector4 }* 12: } 13: */* SCMA Encoding */* 14: **for** *(j=0; j<6; j=j+1){* 15: **for** *(i=0; i<init_num; i=i+2) {* 16: *En_in[j][0.1] = input_user[j][i. i+1];* 17: *SCMA_En (En_in[j], CB [j], CW_U[j]);* 18: /* PRE Mapping */ 19: **for** *(k=0; k<4 ; k++) {* 20: *PRE[i/2][k] =* $\sum_{n=1}^{6} CW\_U[n][k]$*; }* 21: }}

In order to leverage this opportunity, the codebook set should be re-arranged and resized to be able to map four bits to two codewords at each iteration. Therefore, for that second software implementation, the codebook set was organized to allow processing four bits at a time instead of two. Table 3.5 shows the related pseudo for this implementation. Although it is expected that when the number of bits processed at each iteration doubles, the processing time per bit could be reduced by a factor of two. However, the observed processing time per bit for the second software

implementation was not halved. This partly comes from an increased complexity when the codebooks are rearranged (See Table 4.1, Chapter 4).

Table 3.5: SCMA Encoding pseudo code for the second implementation

SCMA Encoding pseudo code for second implementation
1: /* SCMA Encoding Function */
2: **void** *SCMA_En (char in_bit[16][4], double _Complex in_cb[8][16], double _Complex out_cw1[16][4], double _Complex out_cw2[4]){*
3: **for** *(i=0; i<16, i++) {*
4: **switch** *(in_bit[i][4]) {*
5: **case** *[i]:*
6: *mapping codeword vector out_cw1[i]*
7: *mapping codeword vector out_cw2[i]*
8: *break*;
9: }}

Another approach to improve processing time is adding more parallelism to the code. In order to do that, the code was rewritten to a configurable code. This version can be configured with different levels of parallelism of 1,2,3 and 6. It means that for example if parallelism is set to 1, two bits of each user are processed per iteration and if parallelism is set to 2, four bits of each user are processed per iteration. The levels of parallelism used in this design were chosen regarding the number of data layers/users. Since then, the levels of parallelism should be dividable by 6 and so the proper levels can be 1,2,3 and 6. Table 3.6 reports the Configurable SCMA Encoding pseudo code. (See Table 4.1, Chapter 4).

The observed results show that throughput does not increase linearly and that there is no further significant improvement if the parallelism is increased beyond 6. Thus, when the code is instantiated with a parallelism of 6, that basic algorithm saturates the hardware resources available to a single thread on an i7. Therefore, a future approach could be using pthreads to increase processing speed by leveraging the parallelism available in the algorithm.

Table 3.6: Configurable SCMA Encoding pseudo code

Configurable SCMA Encoding pseudo code
1: **void** *SCMA_Encoder (char input_user1[2*PARALLELISM], char input_user2[2*PARALLELISM], char input_user3[2*PARALLELISM], char input_user4[2*PARALLELISM], char input_user5[2*PARALLELISM], char input_user6[2*PARALLELISM], double _Complex PRE[PARALLELISM][4])*
2:{
3: /* SCMA Encoding */
4: **for** *(j=0; j<6; j=j+1){*
5: **for** *(i = 0; i < PARALLELISM; i++) {*
6: *En_in[j][0.1] = input_user[j][2*i. 2*i+1];*
7: *SCMA_En (En_in[j], CB[j], CW_U[j]);*
8: /* PRE Mapping */
9: **for** *(k=0; k<4 ; k++) {*
10: *PRE[i][k]* = $\sum_{n=1}^{6} CW\_U[n][k];$ }
11: }}

There is still some room to do more and decrease the processing time per bit and consequently increase the throughput. One option in order to improve SCMA encoder design is characterizing the complexity of SCMA encoding algorithm and reducing its computational complexity to achieve lower processing time per bit and thus higher throughput. In addition, by increasing the number of data layers in each SCMA encoder and/or the number of parallel SCMA encoders, the number of bits processed at each iteration would increase.

3.3.1 Improved SCMA Encoder Software Implementation and Results

In order to assess the complexity of SCMA encoding and multiplexing for PRE mapping, software implementations were developed with the C programming language. These implementations assume six data layers/users and four physical resources. Again the same compatible sample codebook set was used [52].

In the previously discussed implementations, a configurable SCMA encoder with different levels of parallelism has the best results. As the number of data layers/users is six, four levels of parallelism were developed in this design. Since parallelism values determine the number of bits per user processed in each function call, the number of bits of each data layer/user processed per iteration is equal to the parallelism value times two. For example, if the parallelism value is set to 1, two bits of each data layer/user are processed per iteration and for a parallelism value of 2, four bits of each data layer/user are processed per iteration. Besides and in the first attempt, in order to reduce the computational complexity, the short integer data type was used for inputs and outputs. In general, using short integers instead of floating point numbers has an impact on accuracy. But in SCMA encoding implementation, using short integer instead of floating point has no impact on accuracy because the inherent feature of codewords elements. The reason is that the codewords values used in the implementations have zero integer value and have just four digits in their mantissa part. So, they can totally fit into short integer without any rounding up. Also in this implementation, fixed point can be used with so many bit without any influence on computation accuracy, we just need to put the fixed decimal point after fifth digit of codwords elements value and scale back the results accordingly.

Although the results of that implementation were good, as will be shown later, they did not meet target goal to have more than 1Gbps throughput. Therefore the SCMA encoder was redesigned and some improvements were done in order to reduce SCMA encoding processing time (Latency) and consequently increase its throughput.

For the next approach, the inherent feature of SCMA encoding algorithm was considered. In designed SCMA encoder implementations, there are four PREs that transmit codewords of six data layers/users and for this purpose some adders are normally used. As the user codebooks are constant, the results of the adders, which are the values transmitted over PREs, are predictable and can be computed in advance. Therefore, in order to reduce the computation time in the SCMA encoder, the PRE adders are replaced with a pre-computed look up table. This design is called in the rest of this section the Configurable SCMA Encoder with No Adder. Actually, for each bit of result five adds should be performed so an SCMA encoder with no adder saves 5 additions. In that case, look up tables need to be pre-computed in advance. A C code implementation leveraging such a table was produced. The size of the table is determined by the number of data layer/users and

number of PREs. The No adder version of Configurable SCMA encoder pseudo code to is reported in Table 3.7.

Table 3.7: Configurable SCMA Encoding with No Adder (Multiple Table) pseudo code

Configurable SCMA Encoding with No Adder (Multiple Table) pseudo code

1: /* SCMA Encoding - PRE mapping Function */
2: **extern const short _Complex** *CB[PARALLELISM][4096][4];*
3: **void** *SCMA_Encoder (unsigned int received_data [PARALLELISM], short _Complex PRE [PARALLELISM] [4])*
4: {
5: **for** *(i = 0; i < PARALLELISM; i++)* {
6: **for** *(k=0; k<4 ; k++)* {
7: *PRE[i][k] = CB[i][received_data[i]][k];* }
8: }}

Besides, to further reduce complexity, the short integer data type is used for inputs and outputs instead of the floating point data type. In this design, just one pre-computed table is used to produce four PREs output values. In order to further leverage data layers/users independence and experiment more on exploitable parallelism, instead of using one pre-computed look up table, multiple tables were used to generate partitioned outputs. In this manner since there are four PREs in the designed SCMA encoder, four look up tables can be used. Again these four look up tables should be generated and available for the main design. Incomming data in this implementation shown as *received-data* in following pseudo code is the concatenation of the bits from all users and consequently the size of *received-data* is 12bits. The size of each table is going to be 4096 complex values. This comes from the number of data layer/users and number of PREs existing in this design. Therefore, significant improvements in processing time (1/throughput) and throughput were obtained by using multiple tables (See Table 4.2, Chapter 4).

Although a throughput of more than 1Gbps is gained, the results are somewhat disappointing as gains are sub-linear. It seems that there is no reasonable logical relation between

the performances with different parallelism values but the reason for achieving these results is that the algorithm has enough intrinsic parallelism but the hardware on which it executes cannot exploit it. For instance, when the code is instantiated with a parallelism of 6, the basic algorithm saturates the hardware resources available to a single thread on an i7. One other reason can be cache effect, the size of pre-computed LUTs are 4096 x 4 complex values that may affect L1 cache performance.

After analysing the implementation results for SCMA encoder with no adder, a question was raised regarding how much saving happens in data computation time with no adder design.

In the reported implementations, four PREs was assumed. To compute each PRE output, codewords of different data layers/users should be added together and since in the software implementation these additions are executed sequentially, it takes the maximum time to proceed in comparison with other part. The most time consumption part in SCMA encoder main function for Configurable with Short integer version is SCMA encoding function that consists of PRE adders (See Table 4.3, Chapter 4). Thus, there is a trade-off between live computation and pre-computed look-up tables. However, the PREs results are completely predictable, so when using pre-computed tables, we just leverage the inherent feature of the SCMA encoder algorithm to reduce computation complexity.

If this implementation meets our throughput target, its performance can be further improved by using SIMD operation to unroll the loops and pthreads to further exploit parallelism and make SCMA encoders generate more outputs in parallel.

3.4 Vivado HLS Implementation Experiments

In order to obtain a hardware implementation of the SCMA encoder and PRE mapping, a C based function and testbech were used as inputs to Vivado HLS. Regarding the software implementation, several implementations were done. In the first HLS implementation, at each iteration, two bits per data layer are encoded. As Vivado HLS features were explained in Chapter 2, for each synthesis, some constraints or directives can be added considering the characteristics of the design. For the first solution (Solution A) no user constraints or directives were applied to obtain a reference design, but for the second solution (Solution B) all the loops were unrolled. Unrolling the loops consumes more resources as iterations are executed in parallel but it improves

the processing time. As it is observed in the results, the resource usage for Solution A is less than for Solution B. On the other hand, the processing time (1/throughput) for Solution B was significantly reduced (See Table 4.4, Chapter 4).

Increasing the number of bits encoded per data layer and per iteration decreases the processing time per bit. To achieve better performance in terms of processing time, a second design for HLS implementation (Solution C) was used. This implementation exploits a re-arranged codebook set that processes four bits per users at each iteration. The synthesis was done with unrolled loops and stream type inputs. Unexpectedly, the processing time is not improved in spite of an increased complexity and it is even worse than the results obtained for Solution B. One reason can be higher complexity in this design that causes higher processing time and consequently lower throughput (See Table 4.5, Chapter 4).

For the third HLS implementation (Solution D), the algorithm is changed to support configurable parallelism. The same design with what implemented as Configurable SCMA encoder in software was used for this solution. The best results were achieved when the configurable encoder works with a parallelism of 6 and consequently processes 12 bits per data layer at each iteration. In this implementation, all loops are unrolled and the main function is pipelined. Moreover, the code is more flexible, which allows adding more directives to approach different results (See Table 4.6, Chapter 4).

One of SCMA encoding function features is that the codebook sets used are constant and this feature can play a role to improve the results. Therefore, in solution E, one more directive was added to Solution D that considers this feature. Function-Instantiate was used to leverage constant values in codebook sets (See Table 4.7, Chapter 4).

Solution E delivers better results but not as much as expected. As in this algorithm computes with floating point numbers by default, even if they are not needed, the data type was changed from floating point to short integers, in order to decrease hardware requirements in an FPGA in Solution F. The used directives for Solution F are the same with Solution E. The expectation is to have a smaller latency, higher throughput and a very significant reduction of hardware resources requirements in observed synthesis results (See Table 4.8, Chapter 4).

The best solution reached a throughput of a little bit more than half of the target throughput, while consuming very little resources compared to the first experiments. This demonstrates the

benefits that a tool such as Vivado HLS can provide but the design should be reconsidered in order to improve the performance (See Table 4.9, Chapter 4).

3.4.1 Improved Vivado HLS Implementation

In order to perform high level synthesis of SCMA encoders, like the previous implementations, the C – based function and testbech generated for software implementation were used as inputs to Vivado HLS. The target technology used for these implementations is the Xilinx Virtex-7 FPGA. After analysing the previous obtained results, it was noticed that something happened in generating parallel output. Since for different level of parallelism different synthesis latency was observed, it was figured out that Vivado HLS did not generate the PRE outputs partitioned to perform parallelism regarding parallelism value. The outputs are generated by PRE adders, thus it is needed to find a way to make adders generate output in parallel. Therefore, in order to overcome this obstacle and regarding the software implementation, the No Adder version of the configurable SCMA encoder were synthesized to improve the results.

In this synthesis, in order to help and guide Vivado HLS to generate partitioned PRE outputs, the version with multiple pre-computed tables was used. This design was selected to leverage the existing potential of producing parallel outputs. Synthesis was done by adding some directives to generate better results. These directives include Function Pipeline, Unroll and Function Instantiate; Function Pipeline was applied on the main function and Unroll was applied on all loops. Also, Function Instantiate was applied on the SCMA encoding function with constant values as its inputs to ensure that the tool considers the codebook sets as constants. Furthermore, for Solution G1 of this implementation the clock period was set to 10ns. During synthesis and performance analysis, we noticed that synthesis could be executed with smaller clock periods as well. Therefore, two more different clock periods, 5ns and 2.5ns in Solution G2 and Solution G3 were applied respectively. With the same set of directives, the latency and throughput are the same for different levels of parallelism. It means that with this set of directives the tool does not understand that it should generate parallel outputs. Therefore, it is noticed this time that the directives used need to be modified. Besides, it is figured out that the synthesis latency is not the proper metric for processing time. Therefore, the term “Interval” is used as the processing time metric. *Interval* is the time duration before the function initiates a new set of input and starts to process the next set of input data. Therefore, regarding the definition of Interval, when the Function

Pipeline directive is used, *Interval* is a more reliable metric than *Latency* from high level synthesis report [51] [54] (See Table 4.10, Chapter 4).

By analyzing the results and SCMA encoder performance, it became clear that the directive set used was not suitable to make the tool produce outputs in parallel, thus the outputs are not generated in parallel. Moreover, after more analysis, it was noticed that the unrolled loop inside the pipelined function is ignored by the tool. Therefore, using the unroll directive with this implementation does not improve the SCMA encoder performance and can be eliminated.

In order to overcome this obstacle, it became obvious that another directive set should be experimented. This directive set includes Function Pipeline, Function Instantiate and Array Partition. The *Array Partition* directive leads the synthesis to generate partitioned outputs. The new set of directives was applied to two different versions of our SCMA encoder implementation, the Configurable SCMA encoder with Parallelism and short integer data type and the Configurable SCMA encoder with No Adder and short integer data type. For both versions, synthesis was done with four different parallelism values. Solutions H1, H2, H3 and H4 are the synthesis performed for four different parallelism values. After analysis among different synthesis with different clock constraints, the clock period of 5ns was selected for these solutions. The results (See Table 4.11, Chapter 4) show that the obtained intervals for four different parallelism values are the same and it means that for processing two bits, four bits, six bits and twelve bits at each iteration the latency remain constant for these solutions. Thus, it indicates that Array Partition directive made the tool generate outputs in parallel. As was explained in the previous subsection, the parallelism value determines the number of bit per users processed at each iteration. Therefore, the processing time per bit (1/throughput) for higher parallelism value when the interval is constant is decreased.

As mentioned above, synthesis of the Configurable SCMA encoder with No Adder was also re-done with the new set of directives, Function pipeline, Function Instantiate and Array Partition. Since this version of the SCMA encoder has less computational complexity, this implementation has even better performance in terms of processing time (1/throughput) and throughput in comparison with the previous one (Solution H1, H2, H3 and H4) (See Table 4.12, Chapter 4).

As in Configurable SCMA encoder with No Adder implementation, pre-computed tables were used and tables were stored in BRAM_18Ks, LUT were not used in Solution I1 and Solution

I2 because all of values needed are available in BRAM_18K. Note that the tool reports the resource utilization results and tool usually optimizes the synthesis in terms of resource utilization. Also note that the SCMA encoder is a very simple module with no control logic. Actually, the no adder version is just a group of memory so, since there is no address decoding, then no logic is needed.

While there is a need to make comparison among different designs applying Area×Time analysis provide a suitable factor to find the best results. For the Area×Time analysis, Time was taken as ns/bit and Area was computed as average utilization of each resource category using following equation:

$$\text{Area} = \left(\frac{\text{used BRAM\_18K}}{\text{available BRAM\_18K}} + \frac{\text{used DSP48}}{\text{available DSP48}} + \frac{\text{used FF}}{\text{available FF}} + \frac{\text{used LUT}}{\text{available LUT}}\right) / 4$$

Although the results of applying Area×Time analysis shows that there is not a significant difference between Solution I2 and Solution I4, but Solution I2 has the best results (See Table 4.14, Chapter 4).

3.5 Summary on Efforts to Implement the SCMA Encoder

The model illustrated in Figure 3.3 reports the structure of a MATLAB implementation of a complete SCMA transmitter chain. Although in this simulation a SCMA sample codebook set was used for SCMA encoding, the proposed model can be flexible for other codebooks, and even a dictionary of codebooks if the SCMA encoding function is made sufficiently generic. Different codebooks could be used for SCMA encoding of different numbers of data layers/users and different numbers of physical resources. The next step of this work was to explore the need of a hardware implementation. This was done based on the system constraints and the presented model using Xilinx Vivado HLS. By using the Xilinx Vivado HLS tool, a C specification could be transformed into a register transfer level (RTL) implementation and then it would be synthesizable to Xilinx FPGA. The C specification can be written in C, C++ or SystemC. The FPGA has benefits for some applications in performance, cost and power in comparison with traditional processor and can support a parallel architecture. In addition, high-level synthesis connects software and hardware domains. It helps hardware designer work at a higher level of abstraction while creating high-performance hardware in order to improve productivity for hardware designers. Also, it would be helpful in order to accelerate the computational parts of algorithms on a new compilation target of FPGA.

Several designs of an SCMA encoder were presented in this chapter. They were developed to characterize computational requirements and need for hardware acceleration. These different designs explored how to achieve better performance in terms of processing time. To evaluate the proposed designs and to obtain possible hardware implementations, Vivado HLS was used. Vivado HLS allows managing the design by imposing different constraints to find the best solution before synthesizing hardware. The processing time per bit measured for the different hardware designs were reported in Chapter 4 respectively when executing on Virtex-7.

In addition, in this chapter, improved software implementations of SCMA encoding were presented in section 3.3.1. They were developed to characterize computational requirements and needs for hardware acceleration. These designs are the outcomes of exploring and performance of previous implementations reported in section 3.3. Profiling the software implementations on Intel i7 core produced significant improvements in terms of processing time per bit (1/throughput) and throughput. Again, Vivado HLS was used to evaluate the proposed designs and to obtain possible hardware implementations.

Vivado HLS allows managing the design by imposing different directives and constraints to find the best solution before synthesizing hardware. The target device is the Xilinx Virtex-7 FPGA and execution was done with three different clock periods (Solution G1, G2 and G3). The Vivado HLS synthesis results indicate that the tool does not consider the supposed potential parallelism to generate the outputs. Therefore a new set of directives including Function pipeline, Function instantiate and Array Partition was experimented for the next approaches. By using the new set of directives, Vivado HLS generated the outputs in parallel and the results show the same interval for different parallelism values for Configurable SCMA Encoders with parallelism and short integer data type version (Solution H1, H2, H3 and H4). The synthesis was re-done for the Configurable SCMA Encoder with No Adder that was specified with the short integer data type, as well (Solution I1, I2, I3 and I4).

Implementing the SCMA encoder and performing high level synthesis using Vivado HLS is a proper way to experiment Vivado HLS and its key features. This synthesis method compares with HDL synthesis consumes less time. Thus, design and implement the new algorithms proposed for different purposes will be less time consuming and consequently more cost effective.

Briefly, what was done in this set of implementations for SCMA encoding is starting with

preliminary design. The design got improved by performing some optimization and complexity reduction in order to achieve better performance in terms of computation time and throughput. In this mile, by leveraging Vivado HLS features and using different constraints and directives, different synthesis solutions were implemented. Each solution depending on the design, constraints and directives delivers a specific performance. Finally, among different designs with different performances, the best design is determined by characterizing the performances in terms of processing time, throughput and resource consumption (area size). This characterizing process can lead to the best design according to an area-time complexity analysis.

CHAPTER 4 IMPLEMENTATION RESULTS

In Chapter 3, various SCMA encoder implementation designs were explored and experimented in software and hardware. At first, different designs were implemented in software using C-based code and then the C-based codes were used as inputs to Vivado HLS in order to implement them in hardware. As mentioned in Chapter 3, the target goal is to achieve a throughput of more than 1Gbps that is considered regarding available throughput in 4G LTE networks. This chapter is dedicated to present and discuss the software and hardware implementations results.

4.1 SCMA Transmitter Chain MATLAB Implementation Results

In section 3.1, a simple SCMA transmitter chain was illustrated in Figure 3.3, which includes three modules: the Turbo Encoder, the SCMA Encoder and the PRE Mapper. This SCMA transmitter chain was implemented in MATLAB as a preliminary step. As our assumption in this implementation is to have six data layers/users and four physical resources, six different shapes and colors were used to display constellation points related to each data user. The simulation results are shown in Figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 and Figures 4.7, 4.8 that are the SCMA encoder outputs and the SCMA transmitter chain outputs are illustrated in Figure 4.1 and Figure 4.2 respectively.

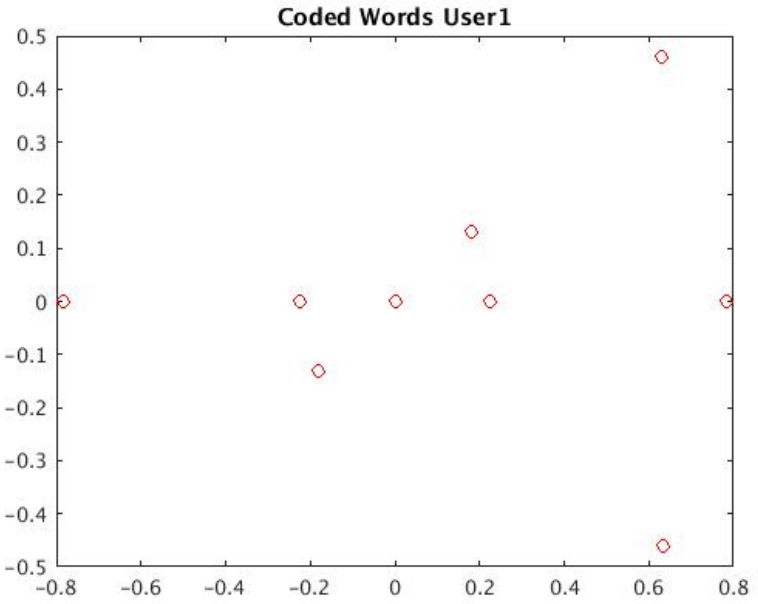

Figure 4.1: SCMA Codewords presented by Constellation points – User1

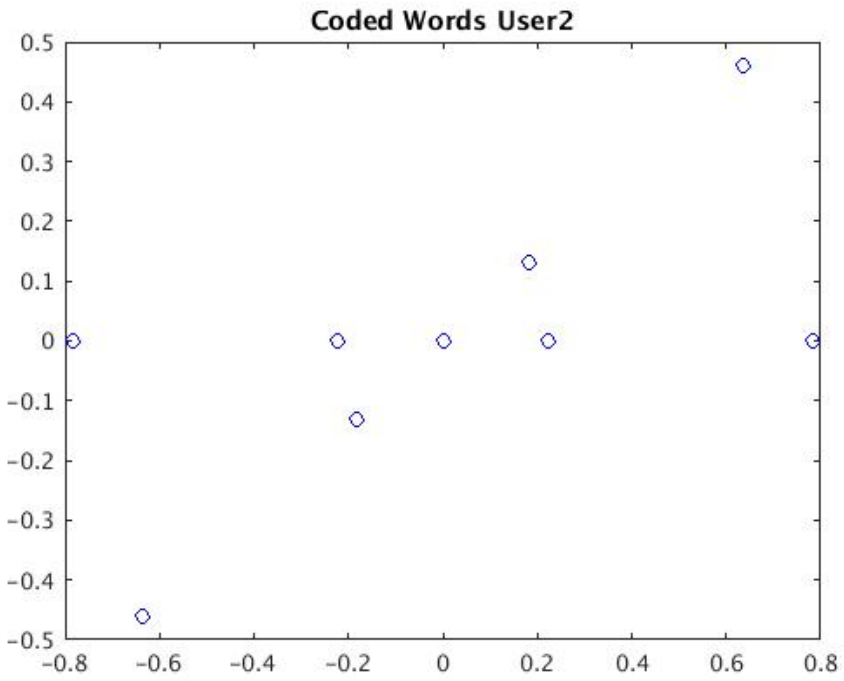

Figure 4.2: SCMA Codewords presented by Constellation points – User2

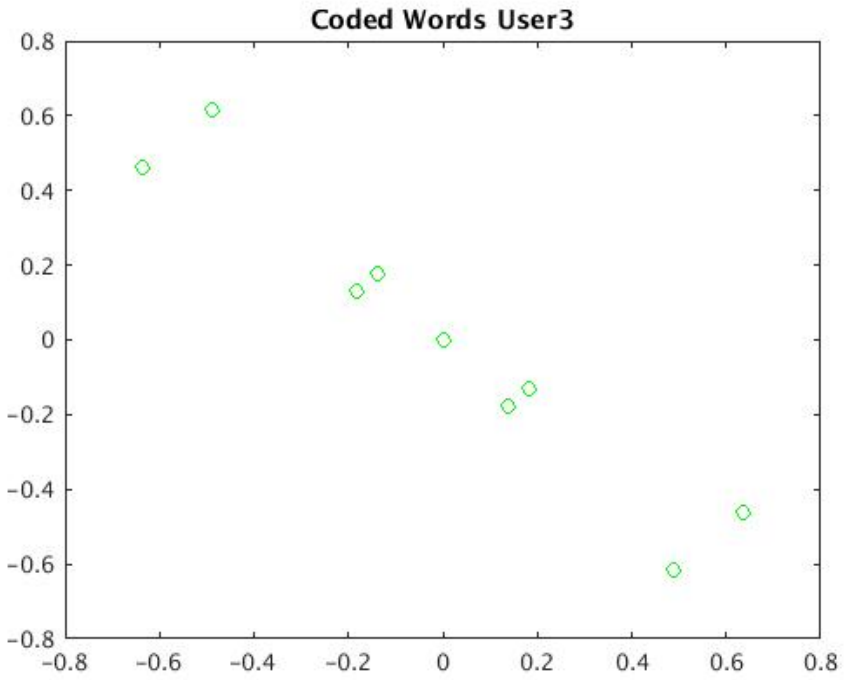

Figure 4.3: SCMA Codewords presented by Constellation points – User3

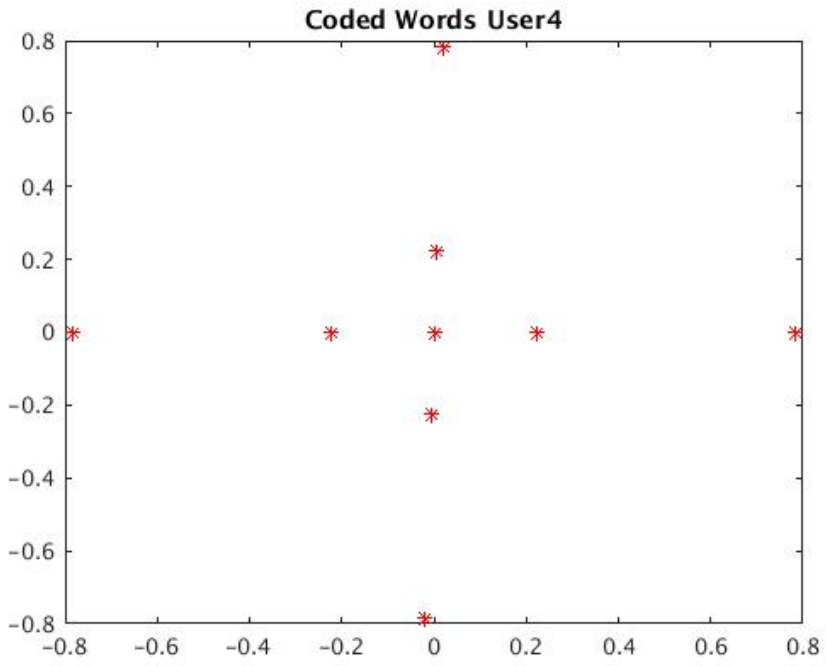

Figure 4.4: SCMA Codewords presented by Constellation points – User4

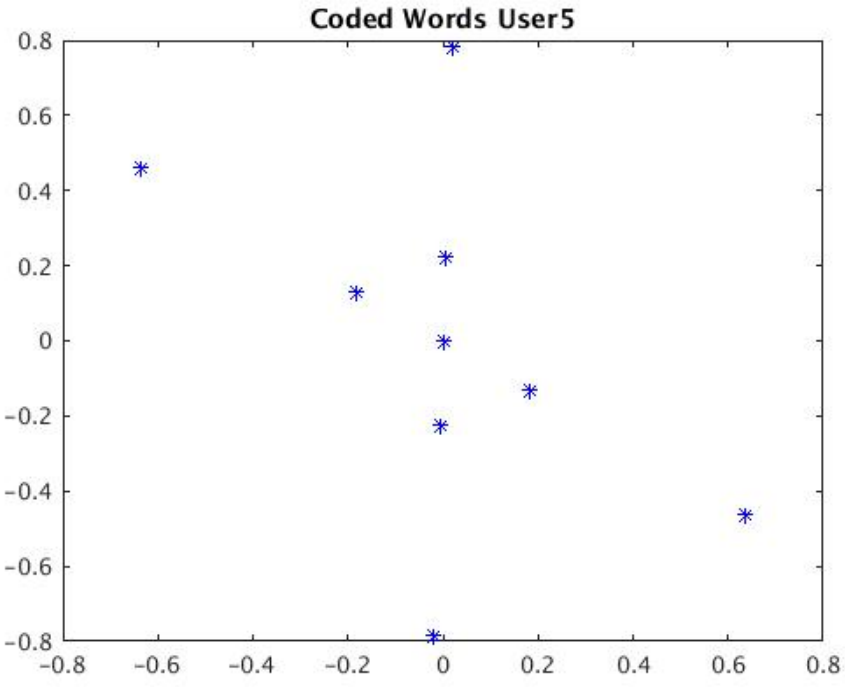

Figure 4.5: SCMA Codewords presented by Constellation points – User5

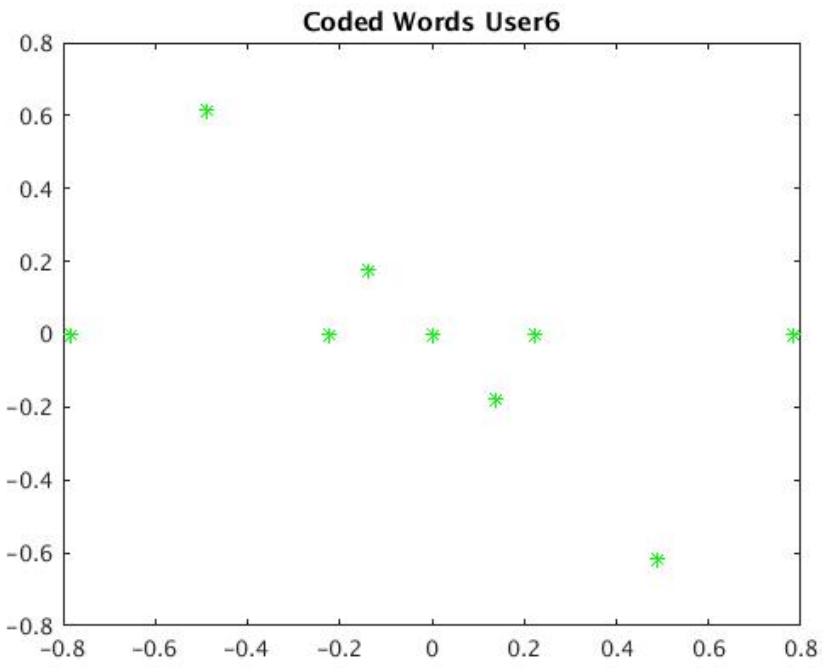

Figure 4.6: SCMA Codewords presented by Constellation points – User6

Different points in Figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 show coded words of different data layers/users. In this implementation there are six data layers/users, so in Figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 there are six different shapes chosen to present constellation points of codewords for different users. There are red, green and blue circles, red, green and blue stars. Each figure shows coded words of one data layers/users. Since some constellation points have the same values, in Figure 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 have some same constellation points.

As explained in chapter 3, section 3.1, each code word should be split into up to four constellation points and each of them should be transmitted over one PRE. Therefore, in order to transmit constellation points of six data layers/users over four PREs the summation of corresponding constellation points of each data layer/user should be transmitted over each PRE.

According to Figure 3.4, each point in Figures 4.7 and 4.8 present the summation of six constellation points of six codewords of different data layers/users. The red round points are related to PRE1, and the blue round points, the green round points and the red star points are presenting the points related to PRE2, PRE3 and PRE4 respectively. As the summation results in some points are the same, some different points are put over each other.

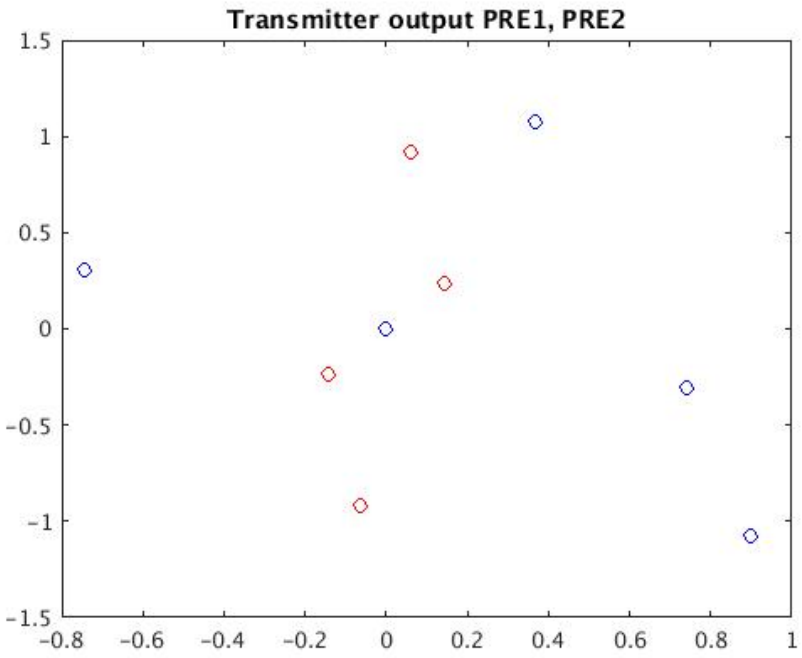

Figure 4.7: SCMA Transmitter outputs presented by Constellation points – PRE1, PRE2

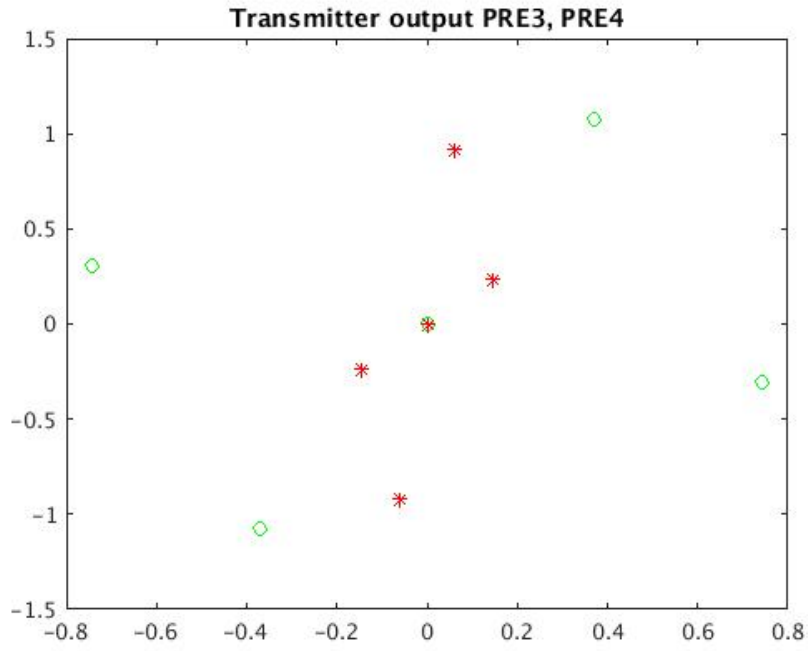

Figure 4.8: SCMA Transmitter outputs presented by Constellation points – PRE3, PRE4

4.2 SCMA Encoder Software Implementation Results

This section is dedicated to present different SCMA software implementations results and discuss about how the observed results were improved through successive implementations. The first software design, as a basic SCMA encoder, encodes two bits of each data layer/user per iteration. In this implementation, the processing time of SCMA encoding and PRE mapping was measured. The processing time of our implementation executing on an Intel i7 3.4GHz Quad-core is about 12.5ns/bit total. As this is far from 1 Gbps, there was a need to improve the processing time.

For the next design, the number of data user bits processed at the same time was increased in order to improve the processing time. Thus, it was expected that when processing two bits at a time, the processing time per bit would decrease by a factor of two. However, the observed processing time per bit for the second software implementation was not halved, as it is about 11.11ns/bit. This partly comes from an increased complexity when the codebooks are rearranged.

For the third approach, different levels of parallelism were added to the algorithm to improve processing time of the code. This version can be configured with different levels of parallelism of 1,2,3 and 6. Implementation results show that the measured processing time for software processing when parallelism is set to 6 is about 7.65ns/bit. The summary of the processing time with these three software implementations are reported in Table 4.1.

Table 4.1: The three preliminary software implementation results summary

SCMA encoder Designs	Processing Time (ns/bit)	Throughput (Mbps)
First	12.5	80
Second	11.11	90
Third (with P=6)	7.56	132.2

This shows that the results did not improve linearly with the parallelism when different levels of parallelism are used, and there is no further significant improvement if the parallelism is increased beyond 6. It was conjectured that there is a saturation of the hardware resource available to a single thread on an i7 when the code is instantiated with a parallelism of 6.

4.2.1 Improved SCMA Encoder Software Implementation Results

The results that will be discussed in this section are related to improved SCMA encoder software implementations. This implementation was explained in section 3.3.1 where it was called a Configurable SCMA Encoder with No Adder and multiple tables. In this implementation, instead of using adders to generate PREs outputs, pre-computed look up tables were utilized that were produced using a piece of C code. Moreover, the short integer data type was used for inputs and outputs instead of the floating point data type to further reduce complexity.

The observed results (Table 4.2) show that a throughput of more than 1Gbps is obtained, which is the assigned target, but the gains are sub-linear with the level of parallelism. One of the probable reasons for the sublinear performance improvement is that when the code is instantiated with a parallelism of 6, the basic algorithm saturates the hardware resources available to a single thread on an i7. One other reason can be a cache effect, the size of pre-computed LUTs are 4096 x 4 complex values that may affect L1 cache performance.

Table 4.2: SCMA Encoder Software Implementation with No Adder and Multiple Table Results

PARALLELISM Value	Processing Time (ns/bit)	Throughput (Gbps)
1	0.55	1.79
2	0.35	2.88
3	0.27	3.59
6	0.5	2.00

Table 4.3: Running time profiling comparison among two versions of SCMA encoder

Running time profiling over 6000000 run (ms)	Main function (ms)	Input initialization (ms)	SCMA encoding function (ms)
Configurable with Short integer	1324 ms	381 ms	759 ms
Configurable with No Adder-Multiple Tables and Short integer	657 ms	358 ms	39 ms

After analysing the implementation results for SCMA encoder with no adder, it became obvious that in a Configurable SCMA encoder with no-adder, 94.8% of the processing time was saved on the SCMA encoding main function that took 720ms less in comparison with the Configurable SCMA encoder with short integer. This amount is obtained by subtracting the amount of time for SCMA encoding function of Configurable SCMA encoder with No adder from the

amount of time for SCMA encoding function of Configurable SCMA encoder with short integer. Table 4.3 shows running time profiling comparisons for two versions of the SCMA encoder called, Configurable with Short integer, and Configurable with No Adder-Multiple tables and Short integer. As it is reported in Table 4.3, significant time is consumed in the main function for the Configurable with Short integer version that uses adders. Running time for this part is about 759ms while this time for the Configurable with No Adder-Multiple tables and Short integer version is about 39ms.

4.3 Vivado HLS Implementations Results

This section is dedicated to report hardware implementation results of different designs that are explained in section 3.4. Each synthesis with Vivado HLS is called a Solution, and some constraints or directives can be added considering the characteristics of the design to each synthesis. Table 4.4 reports the synthesis results of Solution A and Solution B. The table contains processing time, throughput and resource utilization.

Table 4.4: First HLS implementation synthesis results

Synthesis Results		**Solution A Without Directive Constraints**	**Solution B With Directive Constraints (Loop Unrolling)**
Clock (ns)		10	10
Synthesis Latency (clock cycles)		559	34
Latency (ns)		5590	340
1/Throughput (ns/bit)		77.6	4.7
Throughput (Mbps)		12.8	212.7
Utilization (no/%)	**BRAM_18K**	0/0%	0/0%
	DSP48E	6/~0%	48/1%
	FF	30188/3%	15203/1%
	LUT	46033/10%	37648/8%

As Table 4.4 reports, the valuable resource usage like DSP48E for Solution A is less than for Solution B. For example, Solution B uses 48 DSP48E versus 6 in Solution A, but this is about 1% of available DSP48s and it is acceptable. On the other hand, the processing time (1/throughput) for Solution B was significantly reduced. Note that synthesis latency is reported in number of clock cycles. By multiplying the number of clock cycles times the clock period in ns, the latency in ns is calculated. Processing time is obtained by dividing the latency by the number of bits processed in

the implementation. As processing time plays a critical role in real time systems, a solution with lower processing time is usually better if its complexity is acceptable, as is the case with Solution B.

Table 4.5: Second HLS Implementation synthesis results

Synthesis Results		**Solution C With Directive Constraints (Loop Unrolling)**
Clock (ns)		10
Synthesis Latency (clock cycles)		57
Latency (ns)		570
1/Throughput (ns/bit)		7.9
Throughput (Mbps)		126.5
Utilization (no/%)	**BRAM_18K**	0/0%
	DSP48E	96/2%
	FF	25657/2%
	LUT	116640/26%

Solution C exploits a re-arranged codebook set that processes four bits per user at each iteration. The results are reported in Table 4.5 for an implementation with unrolled loops and stream type inputs. As the results in Table 4.5 show, the processing time is not improved, in spite of an increased complexity and this increased complexity can be the reason of higher processing time and consequently lower throughput.

Table 4.6: Solution D HLS implementation synthesis results

Synthesis Results P = 6		**Solution D With Directive Constraints (Loop Unrolling, Function Pipeline)**
Clock (ns)		10
Synthesis Latency (clock cycles)		35
Latency (ns)		350
1/Throughput (ns/bit)		4.8
Throughput (Mbps)		208.3
Utilization (no/%)	**BRAM_18K**	0/0%
	DSP48E	60/1%
	FF	31703/3%
	LUT	33403/7%

Solution D is synthesis of configurable SCMA encoder. The best results were achieved when the configurable encoder works with a parallelism of 6. In this implementation, all loops are unrolled and the main function is pipelined. Table 4.6 summarizes synthesis results. Comparing

Solution D results with Solution B shows no improvement in terms of latency.

Since Solution D is more flexible to add more directives, Solution E can leverage this opportunity by adding one more directive to Solution D. The directive added to Solution D is Function-Instantiate to leverage constant values in codebook sets. Table 4.7 reports the results of recent synthesis. These results show a small improvement in processing time and throughput, but it was not as much as expected. The throughput is still far from 1Gbps.

Table 4.7: Solution E HLS implementation synthesis results

Synthesis Results		**Solution E With Directive Constraints (Loop Unrolling, Function Pipeline, Function Instantiate)**
Clock (ns)		10
Synthesis Latency (clock cycles)		29
Latency (ns)		290
1/Throughput (ns/bit)		4.02
Throughput (Mbps)		248.7
Utilization (no/%)	**BRAM_18K**	0/0%
	DSP48E	63/1%
	FF	22373/2%
	LUT	24687/5%

Table 4.8: Solution F HLS implementation synthesis results

Synthesis Results		**Solution F With Directive Constraints (Loop Unrolling, Function Pipeline, Function Instantiate)**
Clock (ns)		10
Synthesis Latency (clock cycles)		13
Latency (ns)		130
1/Throughput (ns/bit)		1.8
Throughput (Mbps)		555.5
Utilization (no/%)	**BRAM_18K**	0/0%
	DSP48E	0/0%
	FF	670/~0%
	LUT	1364/~0%

Table 4.8 summarizes the synthesis results of Solution F in which a short integer data type was used instead of a floating point data type. These results were obtained with the same directives

as with Solution E. The results of Solution F reported in Table 4.8 are a smaller latency, higher throughput and a very significant reduction of hardware resource requirements.

Table 4.9 summarizes the main results of the high level synthesis experiments reported in this section. The best solution so far reached a throughput of 555 Mbps, while consuming very little resources compared to the first experiments.

Table 4.9: Summary of HLS implementation synthesis results for different implementations

Implementation	No. of SCMA Encoder	No. of processed bits/user/ iteration	Latency (Clock cycles)	Clock period (ns)	Synthesis Latency (ns/bit)	Throughput (Mbps)
B (Loop Unrolling)	1	2	34	10	4.7	212.7
C (Loop Unrolling)	1	4	57	10	7.9	126.5
D (Loop Unrolling, Function Pipeline)	6	12	35	10	4.8	208.3
E (Loop Unrolling, Function Pipeline, Function Instantiate)	6	12	29	10	4.02	248.7
F (Loop Unrolling, Function Pipeline, Function Instantiate)	6	12	13	10	1.8	555.5

4.3.1 Improved Vivado HLS Implementation Results

Table 4.10 summarizes the results of Solution G1, G2 and G3 with parallelism value of 1, but different clock periods. In fact, as Table 4.10 reports, the resource utilization for BRAM_18K and LUT remain the same when the clock period changes (Solution G1, G2 and G3), but the number

of FFs changes. When the clock cycle was pushed, the tool often creates a deeper pipeline, this would increase the number of FFs and it would also increase the "latency" result of the synthesis, but not the "interval" that we use as processing time metric. As it is shown in results reported in Table 4.10, when the opportunity of using smaller clock periods was leveraged, the processing time and throughput were improved and the target throughput was achieved in Solution G2 and G3.

Table 4.10: High Synthesis Results - SCMA Encoder with No Adder and Multiple Tables

Synthesis Results P = 1		**Solution G1**	**Solution G2**	**Solution G3**
Clock (ns)		10	5	2.5
Synthesis Latency (Clock cycles)		3	5	5
Interval (Clock cycles)		2	2	2
Latency (ns)		20	10	5
1/Throughput (ns/bit)		1.67	0.83	0.416
Throughput (Gbps)		0.6	1.205	2.4
Utilization (no/%)	**BRAM_18K**	32/1%	32/1%	32/1%
	DSP48E	0/0%	0/0%	0/0%
	FF	65/0.00008%	159/0.00018%	159/0.00018%
	LUT	74/0.00017%	74/0.00017%	74/0.00017%

For the Configurable SCMA encoder with Parallelism and short integer data type, and the Configurable SCMA encoder with No Adder and short integer data type, synthesis were done with one more directive : *Array Partition*. Table 4.11 reports the results for Configurable SCMA encoder with Parallelism and short integer data type synthesis. Solutions H1, H2, H3 and H4 are the synthesis performed for four different parallelism values.

Table 4.11: High Level Synthesis Results - Configurable SCMA Encoder with Parallelism and Short Integer data type

Synthesis Results		**Solution H1**	**Solution H2**	**Solution H3**	**Solution H4**
Parallelism Value		1	2	3	6
Clock (ns)		5	5	5	5
Interval (Clock cycles)		2	2	2	2
Latency (ns)		10	10	10	10
1/ Throughput (ns/bit)		0.833	0.416	0.278	0.138
Throughput (Gbps)		1.2	2.4	3.6	7.19
Utilization (no/%)	**BRAM_18K**	0/0%	0/0%	0/0%	0/0%
	DSP48E	0/0%	0/0%	0/0%	0/0%
	FF	349/0.04%	665/0.076%	1025/0.118%	1985/0.229%
	LUT	1195/0.275%	1903/0.439%	3095/0.714%	5703/1.316%

Table 4.11 show that the Array Partition directive made the tool generate outputs in parallel and Latency remained the same for different levels of parallelism. Therefore, the processing time per bit (1/throughput) for Solution H4 is minimum, since the interval is constant. The results reported in Table 4.11 show that the target throughput was achieved for Solutions H1, H2, H3 and H4 for a parallelism value of six. Solution H4 offered the best processing time per bit (1/throughput) and consequently the best throughput was achieved. In addition, resource consumption is extremely low even for a parallelism value of six. Specially, no BRAM_18K and no DSP48E were utilized in these solutions. It means that we could match the Intel i7 core performance with a very small amount of FPGA resources.

Table 4.12 reports the results of synthesis of Configurable SCMA encoder with No Adder with the new set of directives, Function pipeline, Function Instantiate and Array Partition, and for different values of parallelism in Solution I1, I2, I3 and I4. This implementation has even better performance in terms of processing time (1/throughput) and throughput in comparison with the previous one (Solution H1, H2, H3 and H4). However, the number of BRAM_18K was increased.

Table 4.12: High Level Synthesis Results - Configurable SCMA Encoder with No Adder and Short Integer data type

Synthesis Results		**Solution I1**	**Solution I2**	**Solution I3**	**Solution I4**
Parallelism Value		1	2	3	6
Clock (ns)		3	3	3	3
Interval (Clock cycles)		1	1	2	3
Latency (ns)		3	3	6	9
1/ Throughput (ns/bit)		0.250	0.125	0.167	0.125
Throughput (Gbps)		4	8	6	8
Utilization (no/%)	**BRAM_18K**	32/1%	32/1%	32/1%	32/1%
	DSP48E	0/0%	0/0%	0/0%	0/0%
	FF	125/0.014%	247/0.028%	248/0.028%	249/0.028%
	LUT	0/0%	0/0%	210/0.024%	435/0.1%

The results represented in Table 4.12 indicate that the Interval for parallelism value of one and two are the same, but for the parallelism value of three and six, the Interval is increased by one and two clock cycles. This larger value happened because BRAM_18K modules are dual port RAMs. Therefore, access to BRAM_18Ks containing per-computed tables is limited when the level of parallelism is more than two. Although this limitation causes increases of the Interval in

Solutions I3 and I4, the achieved throughput is significantly improved to 8Gbps, that is eight times more than what the design was targeted for.

In Virtex-7, which is the target device, the BRAM size is 18Kbits. In the discussed implementations, what is needed is a multiple of 16kbits, thus wasting 2 out 18 bits in the BlockRAMs. For example, in the Configurable SCMA encoder with No Adder (Solution I1, I2, I3 and I4), implementations use 32×16kbit RAM and waste 32×2kbit. Therefore, the RAM overhead in these implementations is small. Table 4.13 reports the summary of memory usage for last two designs.

Table 4.13: Summary of memory usage

SCMA Encoder Implementation	**Memory (kbit)**	
	Available	**Used**
Configurable SCMA (Solution H1, H2, H3 and H4)	52920	0
Configurable SCMA with No Adder (Solution I1, I2, I3 and I4)	52920	512

Considering results reported in Table 4.12, it seems that Solution I2 and Solution I4 have the same performance in terms of latency and throughput but applying Area×Time analysis provide a suitable factor to find the best results among different designs. Table 4.12 shows the results of an Area×Time analysis. Although the results of applying this Area×Time analysis reported in Table 4.14 show that there is not a significant difference between Solution I2 and Solution I4, Solution I2 still has the best results. Solution I4 and I2 have the best performance in terms of processing time per bit and Area×Time analysis respectively, but for those designs, there is a need to generate tables in advance. However, Solution H4 performance is good enough without any need to do more computations.

Table 4.14: Area×Time analysis results

AREA × TIME (ns/bit)	**Solution H1**	**Solution H2**	**Solution H3**	**Solution H4**	**Solution I1**	**Solution I2**	**Solution I3**	**Solution I4**
	0.00066	0.00054	0.00058	0.00053	0.00063	0.00032	0.00044	0.00035

4.4 Summary on the SCMA Encoder Implementations Results

In this chapter, the results of SCMA encoding software and hardware implementations were reported. It was shown that the software implementation of Configurable SCMA encoder with No Adder and multiple pre-computed tables can work at more than 1Gbps throughput that was the target, however the results are disappointing in terms of parallelism. To evaluate the proposed designs and to obtain possible hardware implementations, Vivado HLS was used.

The target device in HLS synthesis is the Xilinx Virtex-7 FPGA. In Solution G1, G2 and G3 execution was done with three different clock periods. The Vivado HLS synthesis results of Solution G1, G2 and G3 indicate that the tool does not generate the outputs in parallel. The synthesis results of Solution H1, H2, H3 and H4 illustrated that by using a suitable set of directives, Vivado HLS generated the outputs in parallel, and the results show the same interval for different parallelism values for Configurable SCMA Encoders with parallelism and short integer data type version. Consequently, the lowest processing time per bit (1/throughput) of 0.138ns/bit and the highest throughput of 7.19Gbps for parallelism value of six was achieved for this design while no BRAM_18K and no DSP48 were consumed.

Solution I1, I2, I3 and I4 synthesis results show that, because BRAM_18K are dual port that causes a port-access limitation for Solution I3 and Solution I4, the intervals for different parallelism values are not the same. However, Solution I2 and I4 produce the best results in terms of latency and throughput that are 0.125ns/bit and 8Gbps respectively.

There is another option to benchmark SCMA encoder implementations performance that is done with what is exist in LTE (Long Term Evaluation) in terms of throughput. In LTE (Long Term Evaluation), for a 20MHz channel, there are 16800 Symbols per ms transmitting a net bandwidth of 16.8Msps. Assuming 64 QAM (6bits per symbol) the throughput is 100.8Mbps for a single chain. This throughput will be four times higher for a LTE with 4x4 MIMO (403.2Mbps). The achieved throughput for the proposed SCMA encoder design is about 20 times higher than the one required by each LTE channel. Another approach to characterize the performance of the SCMA encoder HLS implementation is to compare it with the Xilinx LogiCORE 3GPP LTE TurboEncoder. This TurboEncoder has a processing time of 4.4ns per bit [8] that is significantly more than what the SCMA encoder implementation offers.

CHAPTER 5 CONCLUSION

5.1 Summary of the Work and Contribution

In this thesis, several implementations of an SCMA encoding system were presented as a new uplink multiple access technique for the next generation of mobile networks. As it is predicted that the next generation mobile networks will be deployed in 2020, the wireless networks need significant changes and upgrades in order to meet 5G networks requirements.

Regarding high and growing demands for connectivity, providing massive connectivity with low latency in data delivery is one of the key features of fifth generation of mobile networks. SCMA was proposed as a new signaling method and to enhance 5G with multiple access technology in order to have massive connectivity. On the transmitter side, the sparsity and relatively low complexity of SCMA encoding algorithm allows to obtain high quality encoding systems. In addition, using the non-orthogonal property of SCMA enables to serve more users compared to the OFDMA technology.

Experiments with the SCMA encoding system started with MATLAB implementations and it continued with software implementation using the C language that were finally followed by hardware implementations. The target in the software and hardware implementations was to reduce processing time. Therefore, different versions of those implementations were performed to reduce processing time and to improve throughput for each data layer/user. Although a sample parameter and codebook sets are considered in these implementations, the proposed model is flexible for other codebooks, and even a dictionary of codebooks if the SCMA encoding function is made sufficiently generic. Different codebooks could be used for SCMA encoding. Using more complex codebooks to encode a larger amount of data at the same time can produce better error performance as well as lower processing times and latency.

In Chapter 3, possible implementations were described. In section 3.1, a transmitter chain model with three different modules was explained and then in section 3.2, these models were simulated using the MATLAB platform. In this model in order to implement turbo encoder module, the MATLAB communication toolbox was used. The implementation results for the MATLAB preliminary implementation were discussed in section 4.1, Chapter 4.

In section 3.3, different software implementations were described. The three first implementations in this section were preliminary. The first one did not leverage parallelism and it was used as a reference design to characterize the complexity of different parts of the SCMA encoding algorithm. In the second one, in order to achieve better results in term of processing time, the codebook sets were re-arranged in order to process more incoming data bits in one iteration with the SCMA encoder. Although this implementation was supposed to process incoming data two times faster than the previous implementation, the simulation results did not show this amount of performance improvement. In the third software implementation, since there are six data layers/users in the proposed model, four different levels of parallelism were applied in order to decrease the processing time per bit in the SCMA encoding system. The results for this implementation produced very good improvements in processing time, but the improvement were not linear with parallelism. Simulation results for these three software implementation were reported in section 4.2.

Regarding profiling results and in order to reduce complexity of the algorithm, in section 3.3.1, improved software implementations were described. Considering the characteristics of codewords in different codebook sets used in the proposed model in this thesis, there is a possibility to use the short integer data type instead of the floating point type, which helps to reduce complexity in computation. In addition, this data type changing did not have significant influence on the encoding accuracy. Another approach in this section is to further reduce computational complexity by eliminating adders while encoded data is transmitted over physical resources. In order to do that, and considering the SCMA encoding system features, a pre-computed look up table was used instead of adders in the PRE modules. Then for better results, instead of one pre-computed look up table, four pre-computed look up tables were used. The simulation results for these implementations were discussed in section 4.2.1, Chapter 4 shows that a high performance SCMA encoding with throughput of more than 1Gbps could be obtained. That was the baseline target to achieve for the SCMA encoding system.

Based on different implementations mentioned in section 3.3, Chapter 3 characterized computational requirements and needs for hardware acceleration. The designs in section 3.3.1 are the outcomes of exploring and performance of previous implementations reported in section 3.3. Profiling the software implementations running on an Intel i7 core produced significant improvements in terms of processing time per bit (Latency) and throughput. It was shown that the

Configurable SCMA encoder with No Adder and multiple pre-computed tables can work at more than 1Gbps throughput, that was our baseline target, however the results were disappointing in terms of parallelism. To evaluate the proposed designs and to obtain possible hardware implementations, Vivado HLS was used and described in section 3.4, Chapter 3.

In section 3.4, several hardware implementations using the Vivado HLS platform were described. For these implementations, Vivado HLS leverage the C-codes resulting from the software implementations as inputs, functions and testbenches. In addition, before synthesizing the hardware, there is an opportunity to impose different directives and constraints in order to obtain better solution with Vivado HLS. Solution A, B and C are the preliminary implementations. The difference between solution A and B results came from imposing a directive, unroll loop, to execute synthesis for solution B. Solution C used the re-arranged codebook sets and because it adds more complexity to the code, the results for this solution is worse than solution B. Solution D is the synthesis of the Configurable SCMA encoder that provides different levels of parallelism. In this solution, loops are unrolled. The best results were achieved when the level of parallelism was set to 6. This level of parallelism makes the encoder process 12 bits of each data layers/users at each iteration. The results for these solutions were discussed in section 4.3. The difference between solution D and E is just using another directive, while synthesizing the Configurable SCMA allowed to leverage the fact that there are constant values in the codebook sets. To improve the results of solution E, the Configurable SCMA with short integer data type was synthesized in solution F with the same directive used before. As results in section 4.3 shows, Solution F produced better results then previous solutions, but it was still not sufficient.

In section 3.4.1, improved hardware implementations were described. The target device used for these synthesis is the Xilinx Virtex-7 FPGA. The results in section 4.3 shows that Vivado HLS does not generate the outputs in parallel as it is supposed to. Therefore, in improved implementations, the No Adder version of SCMA encoding system was synthesized. In solution G1, G2 and G3, this version was implemented with different clock periods and the same directives. The goal of using different clock periods is pushing the tool as much as possible to do the processing faster. By analyzing the results in terms of processing time and throughput in section 4.3.1, it has been clear that in this version, the tool did not generate the outputs in parallel as well. Therefore, another directive, Array Partition, was imposed. This directive was applied to two versions, the Configurable SCMA with short integer data type and the Configurable SCMA with

No Adder. Solutions H1, H2, H3 and H4, and solutions I1, I2, I3 and I4 are presented and their synthesises results were discussed in section 4.3.1.

Consequently, for solution H1, H2, H3 and H4 the lowest processing time per bit (Latency) of 0.138ns/bit and the highest throughput of 7.19Gbps for parallelism value of six were achieved for this design, while no BRAM_18K and no DSP48 were consumed. In solutions I1, I2, I3 and I4, because BRAM_18K are dual port that causes port-accessing limitation for Solution I3 and Solution I4, the intervals for different parallelism values are not the same. However, Solution I2 and I4 produce the best results in terms of latency and throughput that are 0.125ns/bit and 8Gbps respectively.

Finally, in order to profile the results and find the best one in terms of memory usage and Area×Time an analysis was performed to show that solutions I2 and I4 provide the best results.

5.2 Future work Objectives

The main concern in this project was designing new accelerators and energy efficient architectures in order to enable Wireless Access Virtualization (WAV), an appropriate way to do so is leveraging high level synthesis. One of the major goals of implementing the SCMA encoder and of performing high level synthesis is experimenting with Vivado HLS to leverage its key features to design and implement new algorithms.

In software implementations of a Configurable SCMA encoder, although applying different levels of parallelism provide different results in terms of processing time per bit, the results did not follow a logical pattern. Since the simulation was run without managing the CPU cores execution, one approach to manage the code execution and threads over different cores of CPU and also profiling the results in a proper manner is using Intel DPDK library. Another approach to make parallelism in SCMA encoder is using the pthread library in software implementations. These approaches can help SCMA encoding implementation to improve the results in terms of processing time per bit and consequently increasing the throughput.

In order to have a proper vision and make a fair comparison between software and high level synthesis implementation, one approach can be leveraging SIMD instructions features to unroll the loops in software implementations. SIMD instructions provide a capability to process at least four input data at the same time in calculations. Therefore, utilizing these features improve

SCMA encoding software implementation by allowing to leverage more parallelism. In SCMA encoding, SIMD instructions can be used. On the other hand, in Configurable SCMA with no adder, the amount of computation power needed for encoding is somehow minimized, and SIMD instructions are less promising.

What is mentioned above is about software implementations future works. In terms of hardware implementation, one of future work objective is benchmarking high level synthesis of SCMA encoding in terms of power and energy consumption per bit for our designs. Since energy consumption per bit is one the criteria for next generation mobile networks, this objective can provide a proper view for SCMA encoding system. In order to measure power consumption, Vivado HLS IP should be extracted and feed to Vivado, then doing a Placement and Routing would enable characterizing power and energy consumption of various implementations. Finally comparing the results with what other encoding systems provide in terms of throughput vs energy consumption per bit would be useful.

BIBLIOGRAPHY

[1] Woon Hau Chin, Zhong Fan, and Russell Haines, “Emerging Technologies and Research Challenges for 5G Wireless Networks,” IEEE Wireless Communications, April 2014

[2] Erik Dahlman, Stefan Parkvall, and John Skold, “LTE/LTE-Advanced for Mobile Broadband”, Academic Press 2011, p1-14

[3] Akhil Gupta, Rakesh Kumar Jha,” A Survey of 5G Network: Architecture and Emerging Technologies,” IEEE Access, August 2015

[4] Shanzhi Chen and Jian Zhao, “The Requirements, Challenges, and Technologies for 5G of Terrestrial Mobile Telecommunication,” IEEE Communications Magazine, May 2014, p 36-43

[5] A. Osseiran et al., “The Foundation of the Mobile and Wireless Communications System for 2020 and Beyond Challenges, Enablers and Technology Solutions,” VTCSpring 2013, June 2–5, 2013.

[6] NGMN Alliance, “Next Generation Mobile Network,” White Paper, February 2015.

[7] Shilpa Talwar et al, “Enabling Technologies and Architectures for 5G Wireless,” IEEE Microwave Symposium , Tampa, pp. 1-4, June 2014

[8] Mohammed Al-Imari, “Uplink Non-Orthogonal Multiple Access for 5G Wireless Networks,” IEEE, 2014, p781-785

[9] Li-Chun Wang et al., “A Survey on Green 5G Cellular Networks,” IEEE, 2012

[10] Saurabh Patel et al., “5G: Future Mobile Technology-Vision 2020,” International Journal of Computer Applications (0975 – 8887) Volume 54– No.17, September 2012

[11] Kelvin Au, Liqing Zhang, Hosein Nikopour, Eric Yi, Alireza Bayesteh, Usa Vilaipornsawai, Jianglei Ma, and Peiying Zhu, “Uplink Contention Based SCMA for 5G Radio Access,” Globecom 2014 Workshop - Emerging Technologies for 5G Wireless Cellular Networks, pp. 900-905, 2014

[12] Zhang S, Xu X, Lu L, et al. “Sparse code multiple access: An energy efficient uplink approach for 5G wireless systems,” in Proc. IEEE Global Communications Conference (GLOBECOM), 2014, pp. 4782-4787.

[13] Y. Grunenberger et al., “Wireless card virtualization: From virtual NICs to virtual MAC

machines," Proc. FutureNetw, Berlin, pp. 4-6, July 2012.

[14] N. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," IEEE Commun. Mag. Volume 47, Issue 7, pp.20- 26, Jul. 2009.

[15] G. Schaffrath et al, "Network virtualization architecture: proposal and initial prototype," Proc. 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures (VISA), pp. 63-72, 2009.

[16] J. Turner and D. Taylor, "Diversifying the Internet," Proc. GLOBECOM '05, vol. 2, pp. 760-766, 2005.

[17] N. Feamster, L. Gao, and J. Rexford, "How to Lease the Internet in your Spare Time," SIGCOMM Comp. Commun. Revi., vol. 37, no. 1, pp. 61–64, 2007

[18] P. Marsch et al, "Future communication networks: challenges in the design and operation," IEEE Vehic. Tehno. Mag., Volume 7, Issue 1, pp. 16-23, Mar. 2012.

[19] S. Perez, J. Cabero, and E. Miguel, "Virtualization of the wireless medium: A simulation-based study," Proc. IEEE 69th VTC-Spring, Barcelona, pp. 1-5, April 2009.

[20] M. Ott, I. Seskar, R. Siraccusa, M. Singh, "ORBIT testbed software architecture: supporting experiments as a service," Tridentcom 2005, pp. 136-145, Feb. 2005

[21] M. Hibler, R. Ricci, L. Stoller, J. Duerig, S. Guruprasad, T. Stack, K. Webb, and J. Lepreau, "Largescale virtualization in the Emulab network testbed," Proceedings of the 2008 USENIX Annual Technical Conference, Boston, MA, pp. 113-128, June 2008.

[22] http://en.wikipedia.org/wiki/Jitter

[23] Yaping Zhu, Rui Zhang-Shen, Sampath Rangarajan, and Jennifer Rexford "Cabernet: connectivity architecture for better network services". In Proceedings of the 2008 ACM CoNEXT Conference, Spain, page 64. ACM, 2008.

[24] Yonghua Lin, Ling Shao, Zhenbo Zhu, Qing Wang, and Ravie K Sabhikhi, "Wireless network cloud: Architecture and system requirements," IBM Journal of Research and Development, Volume 54, Issue 1, pp. 1-12, 2010.

[25] Cisco Systems, "Cisco VN-Link: Virtualization-Aware Networking". White Paper, 2009.

[26] InterDigital. Analysis of feedback mechanisms for CoMP. 2009.

[27] Andy Bavier, Nick Feamster, Mark Huang, Larry Peterson, and Jennifer Rexford, "In vini veritas: realistic and controlled network experimentation," In ACM SIGCOMM Computer Communication Review, volume 36, pp. 3–14. ACM, 2006.

[28] Huawei Technologies, "Consideration on CoMP for LTE-Advanced," 3GPP/R1-083049.

[29] A. Y. Panah et aI., "Utility-Based Radio Link Assignment in Multi-Radio Heterogeneous Networks," IEEE Globecom workshop on LTE and Beyond 4G Technologies, pp. 618-623, Dec. 2012.

[30] Cheng-Xiang Wang et al, "Cellular Architecture and Key Technologies for 5G Wireless Communication Networks," IEEE Communications Magazine, Volume 52, Issue 2, pp. 122-130, February 2014

[31] S. Rangan et aI., "Millimeter wave cellular wireless networks: potentials and challenges," Proceedings of the IEEE, Volume 102, Issue 3, NYU Wireless Center, pp. 366-385, January 2014.

[32] F. Rusek et al., "Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays," IEEE Sig. Proc. Mag., volume 30, Issue 1, pp. 40–60, Jan. 2013

[33] Erik Dahlman et al, "5G WIRELESS ACCESS: REQUIREMENTS AND REALIZATION," IEEE Communications Magazine — Communications Standards Supplement, Volume 52, Issue 12, pp. 42-47, December 2014

[34] https://msdn.microsoft.com/en- us/library/aa925764.aspx

[35] Aaron Zinman and Judith Donath, "Signs: Increasing Expression and Clarity in Instant Messaging," 42nd Hawaii International Conference on System Sciences, MIT Media Lab Sociable Media Group, pp. 1-10, 2009

[36] Parameswaran Ramanathan and Kang G. Shin, "Delivery of Time-Critical Messages, Using a Multiple Copy Approach," ACM Transaction on Computer Systems, Volume 10, Issue 2, pp. 144-166, 1992

[37] Jose F. Monserrat et al, "Rethinking the Mobile and Wireless Network Architecture, The METIS Research into 5G," 2014 European Conference on Networks and Communications,

Bologna, pp. 1- 5, June 2014

[38] Saddam Hossain, “5G Wireless Communication Systems,” American Journal of Engineering Research (AJER), Volume 02, Issue 10, pp. 344-353, 2013

[39] Chih-Lin I et al, “Toward Green and Soft:A 5G Perspective,” IEEE Communications Magazine, Volume 52, Issue 2, pp. 66-73, February 2014

[40] C. M. R. Institute, “C-RAN: The Road Towards Green RAN,” White Paper, Oct. 2011, available: labs.chinamobile.com/cran.

[41] Mahmoud Taherzadeh, Hosein Nikopour, Alireza Bayesteh, and Hadi Baligh, “SCMA Codebook Design,” Ottawa Wireless R&D Centre, Huawei Technologies Canada Co., LTD. in Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th, pp. 1-5, IEEE, 2014.

[42] Hosein Nikopour, Eric Yi, Alireza Bayesteh, Kelvin Au, Mark Hawryluck, Hadi Baligh, and Jianglei Ma, “SCMA for Downlink Multiple Access of 5G Wireless Networks,” Globecom 2014 - Wireless Communications Symposium, pp. 3940-3945, 2014

[43] Hosein Nikopour and Hadi Baligh, “Sparse Code Multiple Access,” in in Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on, pp. 332-336, IEEE, 2013.

[44] Tingting Liu, Xinmin Li, Ling Qiu, “Capacity for Downlink Massive MIMO MU-SCMA System,” Wireless Communications & Signal Processing (WCSP), 2015

[45] D.J. Love, R.W. Heath, V.K.N. Lau, D. Gesbert, and M. Andrews, "An overview of limited feedback in wireless communication systems,” IEEE Journal on Selected Areas in Communications, vol. 26, no. 8, pp. 1341-1365, October 2008.

[46] ICT-317669, “D4.3: Final Report on Network-Level Solutions.” Feb, 2015. [Online] https://www.metis2020.com/documents/deliverables/.

[47] Erik Dahlman, Stefan Parkvall, and Johan Skold, “4G: LTE/LTEAdvanced for Mobile Broadband,” 2nd ed. Waltham, MA, USA: Elsevier, 2014.

[48] R. Hoshyar, F.P. Wathan, R. Tafazolli, “Novel Low-Density Signature for Synchronous CDMA Systems Over AWGN Channel,” IEEE Trans. On Signal Processing, vol. 56, No. 4, pp.

1616-1626, April 2008

[49] D. Guo, C.C. Wang, “Multiuser Detection of Sparsely Spread CDMA,” IEEE J. Selected Areas Communication, Special Issue on Multiuser Detection for Advanced Communication Systems and Networks, vol. 26, pp. 421-431, April 2008

[50] A. Mathur, M. Fujita, E. Clarke, and P. Urard, “Functional equivalence verification tools in high-level synthesis flows,” IEEE Design & Test of Computers , no. 4, pp. 88-95, 2009.

[51] Xilinx, Inc., “Ug871 - vivado design suite tutorial: High level synthesis.” http://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug871-vivado-high-level-synthesis-tutorial.pdf , May 2014.

[52] Huawei, The 1st 5G Algorithm Innovation Competition SCMA. Altera University Program, 2015.

[53] Xilinx, inc., “DS701 – LogiCORE IP 3GPP LTE Turbo Encoder”, V3.1, June 2014

[54] Xilinx, Inc., “UG902 – Vivado Design Suite User Guide : High Level Synthesis”, v2014, May 2014.

[55] Louise H. Crockett, Ross A. Elliot, Martin A. Enderwitz and Robert W. Stewart, “The Zynq Book-Embedded Processing with the ARM® Cortex®-A9 on the Xilinx® Zynq®-7000 All Programmable SoC,” Department of Electronic and Electrical Engineering University of Strathclyde Glasgow, Scotland, UK 1st Edition, Strathclyde Academic Media, July 2014