Bruno de
Pinho Barbosa

Desenvolvimento em VHDL da Camada Física de um Transmissor 4G

Development in VHDL of a 4G Physical Layer Transmitter

Bruno de Pinho Barbosa

Desenvolvimento em VHDL da Camada Física de um Transmissor 4G

Development in VHDL of a 4G Physical Layer Transmitter

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Arnaldo Oliveira e do Doutor Adão Silva, Professores Auxiliares do Departamento de Eletrónica, Telecomuniçaões e Informática da Universidade de Aveiro.

Este trabalho é financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER), através do Programa Operacional Regional de Lisboa (POR LISBOA 2020) e do Programa Operacional Competitividade e Internacionalização (COMPETE 2020) do Portugal 2020 [Projeto 5G com o nº 024539 (POCI-01-0247-FEDER-024539)].

This work is supported by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Lisbon (POR LISBOA 2020) and the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020 framework [Project 5G with Nr. 024539 (POCI-01-0247-FEDER-024539)].

Para a minha família, em especial para os meus pais e irmão.
Um abraço e felicidades.

o júri / the jury

presidente / president	**Professora Doutora Iouliia Skliarova** Professora Auxiliar da Universidade de Aveiro
vogais / examiners committee	**Professor Doutor Arnaldo Silva Rodrigues de Oliveira** Professor Auxiliar da Universidade de Aveiro (Orientador)
	Doutor Luís Manuel de Sousa Pessoa Investigador Sénior do Instituto de Engenharia e Sistemas de Computadores, Tecnologia e Ciência - INESC TEC - do Porto (Arguente Principal)

agradecimentos / acknowledgements

Primeiro que tudo, quero agradecer aos meus pais e ao meu irmão, pelo apoio que me deram durante o meu percurso académico, para poder ter uma oportunidade de um bom futuro de trabalho.

Quero dar os meus agradecimentos aos meus orientadores, Prof. Doutor Arnaldo Oliveira e o Prof. Doutor Adão Silva, pela dedicação, acompanhamento e disponiblidade que me deram, durante a realização deste trabalho.

Agradeço também à Universidade de Aveiro e ao Instituto de Telecomunicações por me acolherem na sua comunidade académica e ao Departamento de Eletrónica, Telecomunicações e Informática pela oportunidade de aprendizagem e condições de formação, essenciais para o meu percurso académico.

Quero deixar um obrigado ao Pedro Moreira, João Milheiro e Bruno Brandão, com quem aprendi coisas que me foram úteis para o desenvolvimento deste trabalho e me ajudaram no meu trabalho no edifício do IT.

Por último, um agradecimento a todos os meus familiares e aos meus amigos e colegas que conheci na universidade de Aveiro. Um muito obrigado pela vossa ajuda e apoio.

First of all, I would like to thank my parents and my brother, for the support they gave me during my academic career, so I could have the opportunity of a good work future.

I want to give my gratitudes to my advisors, Dr. Arnaldo Oliveira and Dr. Adão Silva, for their dedication, their assistance and willingness they gave me through the fulfillment of this work.

I also would like to thank the University of Aveiro and the Instituto de Telecomunicações for welcoming me to their academic community and the Department of Electronics, Telecommunications and Informatic for the learning opportunity and giving conditions for formation, necessary to my academic course.

I like to give my appreciation to Pedro Moreira, João Milheiro and Bruno Brandão, for giving me help vital to the development of my thesis work and helped me in the work on the IT building.

At last, I like to give my thanks to my family, friends and colleagues that I have met in the University of Aveiro. Thanks for your help and support.

Palavras-chave 5G, 5G-NR, 4G, LTE, SC-FDMA, MATLAB, FPGA, Uplink.

Resumo

A tecnologia LTE e LTE-Advanced são *standards* da quarta geração de comunicações móveis atuais, ou 4G. Futuramente, o 5G marca a próxima geração de comunicações móveis, segundo o *standard* 5G-New Radio (5G-NR).

A tecnologia 5G encontra-se numa fase inicial de implementação, sendo que nessa fase uma das suas características fundamentais é o suporte para comunicações 4G, pois muitos dos dispositivos móveis usados atualmente não possuem suporte para comunicações 5G. Este suporte para 4G é tornado possível, se for feita uma implementação onde as redes 4G e 5G se encontrem em coexistência. No futuro, com o aumento do uso de dispositivos móveis com suporte para 5G, haverá uma migração gradual de redes 4G para 5G, libertando os espectros de frequências reservados atualmente para o 4G para serem ocupados pelo 5G.

As transmissões de dados no 4G exigem bastante da capacidade de processamento de todos os sistemas da rede móvel. Para o 5G, as transmissões de dados tem volumes de tráfego e velocidades maiores do que as transmissões de dados 4G, fazendo com que novos sistemas tenham de ser implementados para poder processar maiores quantidades de dados. A implementação em *hardware* da cadeia de transmissão 4G *Uplink*, ao nível da camada física *PHY-Low*, permitirá a otimização de certos processos que um CPU poderia lidar, diminuindo o uso do CPU e o tempo gasto em processamento. O uso de FPGAs torna isto possível, tendo em conta que podem realizar tarefas em paralelo, em modo simultâneo, e fazer processamento digital de sinal.

O objetivo desta dissertação assenta na modelação de um transmissor 4G LTE *Uplink*, ao nível da camada física. Depois, é gerado código VHDL sintetizável a partir do sistema modelado, que eventualmente será implementada em FPGAs.

A modelação do sistema é feito em Simulink, uma ferramenta no *software* do MATLAB, que permite modelar, simular e analisar sistemas num ambiente gráfico e tem aplicações para sistemas de controlo e processamento digital de sinal.

O código VHDL é gerado a partir do HDL Coder, uma outra ferramenta no *software* do MATLAB, que gera Verilog e VHDL sintetizáveis, a partir de funções MATLAB e de modelos Simulink.

Os resultados obtidos dos dados processados pelo sistema são analisados e validados, comparando com os dados de referência obtidos a partir da *toolbox Wireless Waveform Generator* do MATLAB.

Keywords 5G, 5G-NR, 4G, LTE, SC-FDMA, MATLAB, FPGA, Uplink.

Abstract

The LTE and LTE-Advanced technologies are *standards* to the fourth mobile generation, or 4G. The planned successor of this mobile generation is 5G, which will be based on 5G-New Radio (5G-NR) *standard*.

The 5G technology is on an initial phase of deployment. One of its features that are essential in this initial phase is the support for 4G communications, because many of the mobile devices currently in use do not have support for 5G communications. This support is made possible if there is an implementation where 4G and 5G networks both coexist with each other. In the future, with the increasing usage of mobile devices with 5G support, there will be a gradual migration of 4G networks to 5G, releasing frequency spectrums currently reserved for 4G so that those can be occupied by 5G.

The data transmissions in 4G require quite a lot of the processing capacity of all systems within the mobile network. For 5G, the data transmissions, in terms of traffic volume and speed, are larger than 4G transmissions, requiring new systems to be implemented, to allow the processing of larger quantities of data. Implementation in hardware of a 4G Uplink transmission chain, at the physical layer level PHY-Low, will allow the optimization of certain processes that a CPU could handle, reducing CPU usage and time spent on processing. The use of FPGAs makes this possible, as FPGAs can perform parallel tasks simultaneously and perform digital signal processing.

The purpose of this dissertation is the modelling of a 4G LTE Uplink transmitter, at the physical layer level. Then, synthesizable VHDL code is generated from the modeled system, which can be eventually implemented in FPGAs.

The modelling of the system is made in Simulink, a tool inside the MATLAB software, which allows for modelling, simulating and analyzing systems in a graphic environment and has applications in control systems and digital signal processing.

The VHDL code is generated from HDL Coder, another tool in MATLAB software, generating synthesizable Verilog and VHDL code, from the MATLAB functions and Simulink models.

The results obtained of processed data from the system are analyzed and validated, comparing the reference data generated from Wireless Waveform Generator toolbox in MATLAB.

Contents

List of Figures

List of Tables

List of Acronyms

3GPP	Third Generation Partnership Project
ACK	Acknowledgement
AMF	Access and Mobility Management Function
ASIC	Application Specific Integrated Circuits
BBU	Baseband Unit
BPSK	Binary Phase-Shift Keying
CORDIC	Coordinate Rotation Digital Computer
CP	Control Plane
CPRI	Common Public Radio Interface
CPU	Central Process Unit
CSI	Channel State Information
CU	Central Unit
DFT	Discrete Time Fourier
DMRS	Demodulation Reference Signal
DU	Distributed Unit
DwPTS	Download Pilot Time Slot
eNodeB	enhanced Node Base
EPC	Evolved Packet Core
EVM	Error Vector Magnitude
E-UTRAN	Evolved UMTS Terrestrial Radio Access
FDD	Frequency Division Duplex
FFT	Fast Fourier Transform
FPGA	Field-Programmable Gate Array

gNB next Generation Node Base

GP Guard Period

HARQ Hybrid Automatic Repeat Request

HDL Hardware Description Language

IFFT Inverse Fast Fourier Transform

IMT-Advanced International Mobile Telecommunications-Advanced

ISI Inter-Symbol Interference

ITU International Telecommunication Union

LTE Long Term Evolution

MAC Media Access Control

MER Modulation Error Ratio

MME Mobility Management Entity

ng-eNB next generation eNodeB

NR New Radio

OFDM Orthogonal Frequency Division Multiplexing

OFDMA Orthogonal Frequency-Division Multiple Access

PAPR Peak-to-Average Power Ratio

PDCP Packet Data Convergence Protocol

PUSCH Physical Uplink Shared Channel

PUCCH Physical Uplink Control Channel

PRACH Physical Random Access Channel

QAM Quadrature Amplitude Modulation

QPSK Quadrature Phase-Shift Keying

RACH Random Access Channel

RF Radio Frequency

RLC Radio Link Control

RMS Root Mean Square

RRC Radio Resource Control

RRU Remote Radio Unit

SC-FDMA	Single-Carrier Frequency-Division Multiple Access
SDR	Software-Defined Radio
SNR	Signal-to-Noise Ratio
SR	Scheduling Request
SRS	Sounding Reference Signal
TDD	Time Division Duplex
UCI	Uplink Control Information
UE	User equipment
UL-SCH	Uplink Shared Channel
UMTS	Universal Mobile Telecommunications System
UP	User Plane
UPF	User Plane Function
UpPTS	Download Pilot Time Slot
VHDL	Very High Speed Integrated Circuit (VHSIC) HDL
VSA	Vector Signal Analyzer

Chapter 1

Introduction

1.1 Context

Mobile communications have been on constant evolution over time. The emergence of new mobile devices, such as smartphones, which have more hardware capabilites and more advanced operating systems than older and conventional mobile phones, have made people more demanding from mobile communication services. As the techonology on smartphones have been evolving, the mobile services and applications have also been evolving, like video and audio services and web applications. Because of this factors, the development for larger capacity of wireless communications and faster data transmissions for mobile communications had to be established, and so *standards* are imposed to ensure this development followed the requirements needed.

The Long Term Evolution (LTE) and LTE-Advanced *standards* have been developed, following the guidelines of IMT-Advanced. The goals and objectives of this evolved system include higher radio access data rates, improved system capacity and coverage, flexible bandwidth operations, significantly improved spectral efficiency, low latency, reduced operating costs, multi-antenna support, and seamless integration with the Internet and existing mobile communication systems [1]. LTE and LTE-Advanced represent the fourth generation (4G) mobile communication system. As mobile communication technologies went through four generations, from 1G to 4G, the trend will continue. The next standard, which is the fifth-generation (5G) of mobile communication systems, is currently in early stage of deployment and it is expected to bring significant improvements, in relation to 4G systems. In the section

1.2, it is presented more about 4G and its relationship with 5G, as well as showing more about the network and architecture of both 4G and 5G.

1.2 Motivation

The 5G technology is seen as a generation of wireless mobile communications that will include all kinds of services, which not only mobile devices will be able to connect to the network, but also other existing and future wireless devices, providing connectivity to all devices beyond mobile broadband. For this, 5G will open new doors to developing services/applications around non-mobile wireless devices. As 5G technologies are in its early stages of deployment and most mobile devices in current use only have support 4G communications, the initial course of action is to implement a series of 4G networks coexisting with 5G networks, to provide communications for devices of both mobile generations.

The mobile communications are established through a series of wireless data transmissions between a base station and multiple mobile devices, referred to as user equipment (UE) devices. The UE devices need to establish communication to a base station unit, in order to connect to a mobile network. In LTE *standards*, a base station is defined as an enhanced Node Base station, designated as eNodeB. In 5G-NR *standards*, it is called next Generation Node Base station, or gNB. The communication between UEs and base stations are made possible through air interface. The air interface for 4G LTE communications is Evolved UMTS Terrestrial Radio Access Network - E-UTRAN - and for 5G is Next Generation Radio Access Network - NG-RAN.

On the E-UTRAN air interface, the air interface for LTE, a series of eNodeB and UE are connected to each other. An eNodeB is composed by a Remote Radio Unit (RRU) and a Baseband Unit (BBU). The RRU has the antenna component and makes the conversion between digital signal and radio frequency (RF) signal through Digital-to-Analog (D/A) converters for transmission and Analog-to-Digital (A/D) converters for reception. The BBU controls the RRU and is responsible for the digital signal processing, serving as a intermediate between RRU and core network. The BBU and RRU are interconnected through a fronthaul network on a optic fiber link, with Common Public Radio Interface (CPRI). The communication between two eNodeB are made through the X2 interface, while the base stations are connected to Evolved Packet Core (EPC), the core network for the 4G LTE, through S1 interface. The

S1 interface has two plane interface protocols: control plane (CP) and user plane (UP). The S1-CP and S1-UP interfaces are connected to the Mobility Management Entity (MME) and Serving Gateway (SGW) in the EPC network, respectively. The E-UTRAN architecture is illustrated in figure 1.1.

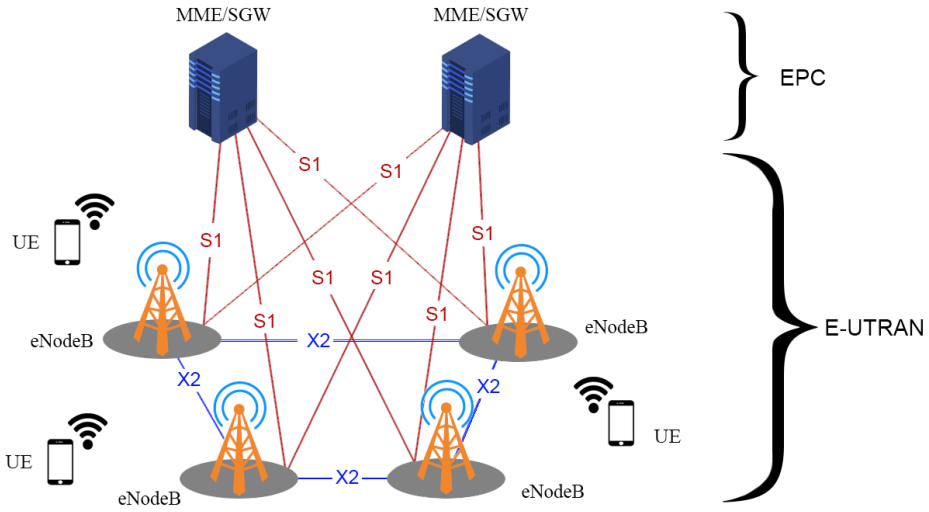

Figure 1.1: E-UTRAN air interface (adapted from [2]).

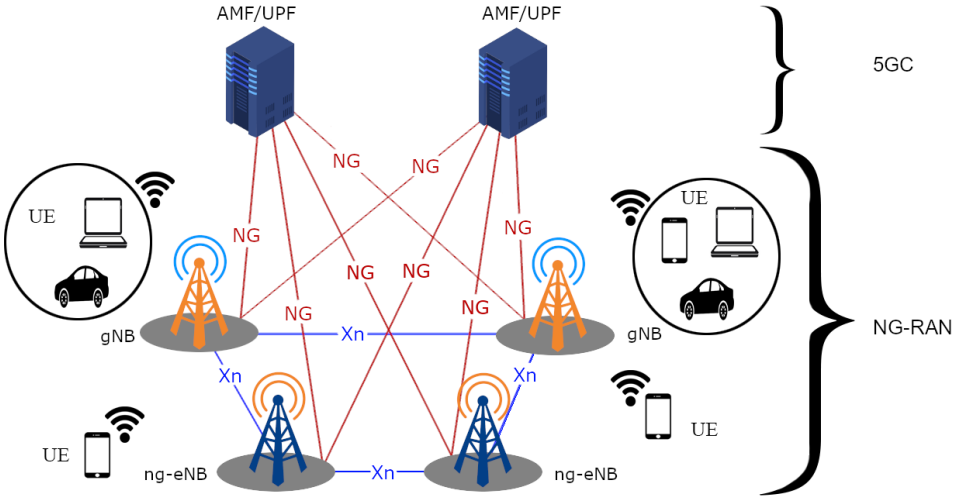

Figure 1.2: NG-RAN air interface (adapted from [3]).

Inside of NG-RAN architecture, there are two types of base station nodes: gNB (as already mentioned) and ng-enB. The latter is a next generation eNodeB, which provides E-UTRAN services to UE, on a 4G-5G network coexistence. Both base stations are interconnected with each other by means of the Xn interface [3] and connect to a 5G Core Network (5GC) through the NG interface. NG-RAN has two plane interface protocols, one CP and one UP, handled on the 5GC by the Access and Mobility Management Function (AMF) and User Plane Function (UPF), respectively. The figure 1.2 shows the NG-RAN air interface architecture.

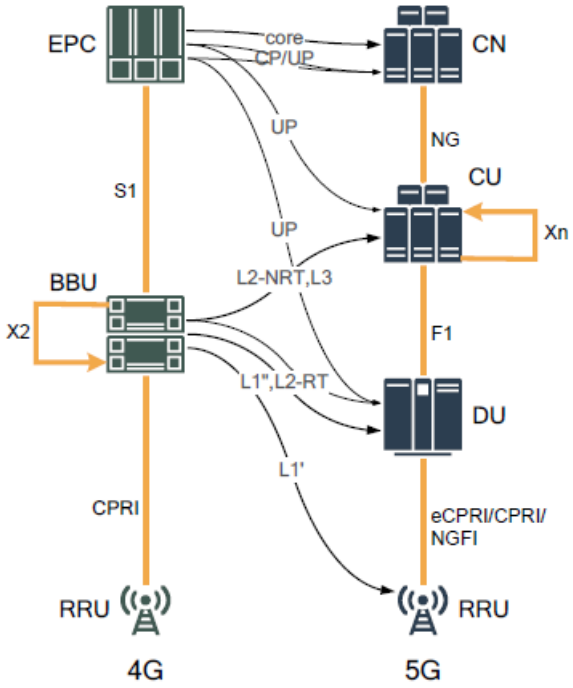

Figure 1.3: RAN evolution from 4G to 5G (extracted from [4]).

From 4G to 5G, in terms of RAN, the BBU functions in 4G are split into three units that composes the gNB: Central Unit (CU), Distributed Unit (DU) and RRU. The EPC has its functions redistributed among the NGC, CU, and DU [4]. Two new fronthaul networks are established, connecting the CU, DU and RRU. Figure 1.3 shows the RAN evolution from 4G to 5G, with yellow lines illustrating transport networks and black lines the function distribution.

The 4G network can coexist with the 5G network, if the 4G and 5G base stations are

interconnected in a non-standalone configuration, shown in figure 1.4. In this coexistence, it is possible for 5G networks to use frequency bands normally used by 4G, dynamically sharing spectrum normally reserved for one of the two types of networks, depending on the UE needs. In the early stage of deployment (option 3), the EPC remains as the core network and is connected to a 4G LTE base station, which is connected to a 5G-NR base station. Later, the 5GC replaces EPC as a core network (option 7), while the 4G LTE base station keeps connection to a core network, with the possibility of connecting to 5GC through a 5G-NR base station (option 4). This evolution will allow the UE (e.g., mobile handsets) to evolve in support of the 5G radio and core functionality [4]. The gradual migration from 4G to 5G allows for 5G networks to operate under the frequency spectrums occupied by 4G, to the point that future deployments of 5G networks work without interventions of 4G networks, acting on a standalone mode.

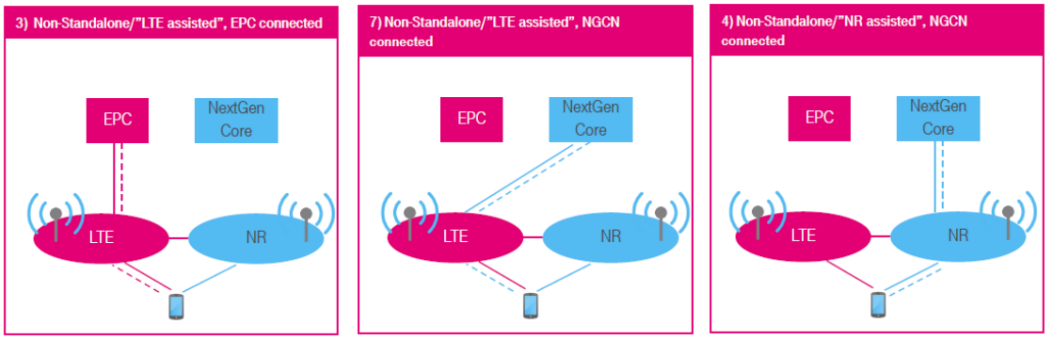

Figure 1.4: Non-standalone configurations (extracted from [4]).

The radio protocol architecture for 4G LTE can be separated into control plane architecture and user plane architecture [5], as well as the 5G architecture. The user plane protocol channels user data, while the control plane protocol ensures connection, setup and security procedures to the bearer for the user plane.

In both the upstream and downstream directions, the radio signals go through a series of signal processing blocks [4]. Figure 1.5 shows these function blocks and split options in both 4G LTE and 5G networks for control and user plane architecture, as well as the mapping of BBU and RRU for 4G architecture and possible mapping options of CU, DU and RU for 5G architecture. The processing blocks are Radio Resource Control (RRC), Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), Media Access Control (MAC), Physical

(PHY) layer and Radio Frequency (RF).

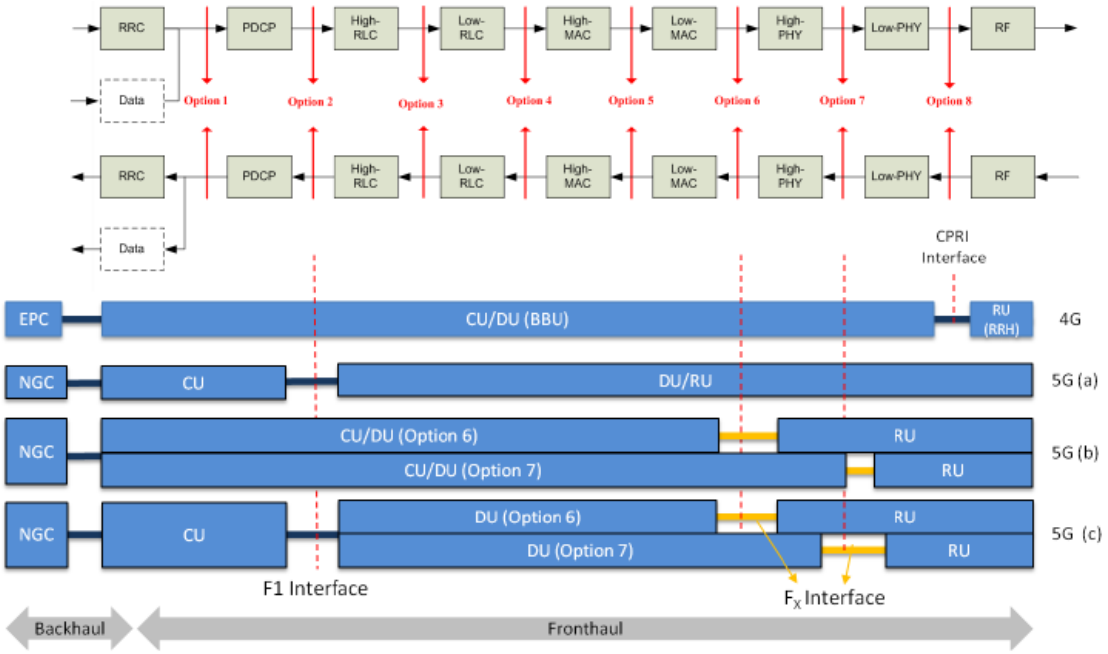

Figure 1.5: Mapping of CU and DU functions according to the split points (extracted from [4]).

The development of a system capable of data transmissions, following 4G LTE or 5G *standards*, has to take into account some factors, such as data transmission rates and the signal processing blocks. The requirements for implementing these processing blocks in software and hardware are higher, as well as data rates, in 5G systems than 4G systems. It is important to implement in hardware the processing blocks with data rates chosen for the system design, to release the software from processing time and improve the system performance. The chosen data rates and processing blocks for the system design is a compromise between the hardware processing capability and the number of processes to be handled by the software.

As seen in figure 1.5, there are several split options which may be implemented in both software and hardware, of which the physical layer options may be implemented in hardware. In a 4G LTE system, the RF and the physical layer block are connected through CPRI fronthaul, with high data rates that would make RF impratical to implement in hardware, because it would occupy the fronthaul bandwidth more than it can handle and create higher latency. For the 5G, the increase in data rates makes it impractical to continue with the conventional CPRI fronthaul implementation. Moving towards a higher layer split would relax the latency

and bandwidth requirements, but then fewer processing functions can be centralized. It is thus critical that the new functional-split architecture takes into account technical and cost-effective trade-offs between throughput, latency and functional centralization [6].

Inside the physical layer processing block, the option 7 functions can be split into options 7.1, 7.2 and 7.3, dividing the processing blocks composing the physical layer processing chain, for downlink and uplink communications. The functional splits at physical layer level are illustrated in figure 1.6. The 7.2 and 7.3 splits not exactly defined, as there are multiple split possibilities for these two splits.

The physical layer chain system, as seen in figure 1.6, involves digital signal processing. For this purpose, Field-Programmable Gate Array (FPGA) hardware can be used to implement the desired physical layer system, since it can perform parallel tasks simultaneously, essential for this kind of processing. The FPGA hardware does not have a defining structure, like the hardware in Central Process Unit (CPU) or Application Specific Integrated Circuits (ASIC), making the FPGA programmable in hardware. This characteristics makes FPGA a viable choice for the development of future communication systems.

In uplink communications, there are two processing schemes: Orthogonal Frequency-Division Multiple Access (OFDMA) and Single-Carrier Frequency-Division Multiple Access (SC-FDMA). The OFDMA processing chain has resource mapping, IFFT and cyclic prefic addition. The SC-FDMA follows the OFDMA processing chain, with an additional process called Transform Precoding, which is a form of Discrete Time Fourier (DFT). This process allows for higher transmission power efficiency, making SC-FDMA scheme more desirable to implement in a design of an uplink transmitter in UE, where transmission power is more limited than in base stations. For this reason, the SC-FDMA scheme is more favorable for hardware implementation in FPGA of an uplink transmitter in UE, than if it used the OFDMA scheme.

The development of a 4G uplink transmitter for UE, in this dissertation, is part of a larger project where the other components, such as downlink transmitter/receiver or uplink receiver, for 4G and 5G, are covered by other aspects of that project. The motivation to develop the uplink transmitter component is the need for testing in controlled environments on a 4G-5G coexistence, in order to validate other system components, from the RF chains to the fronthaul interface.

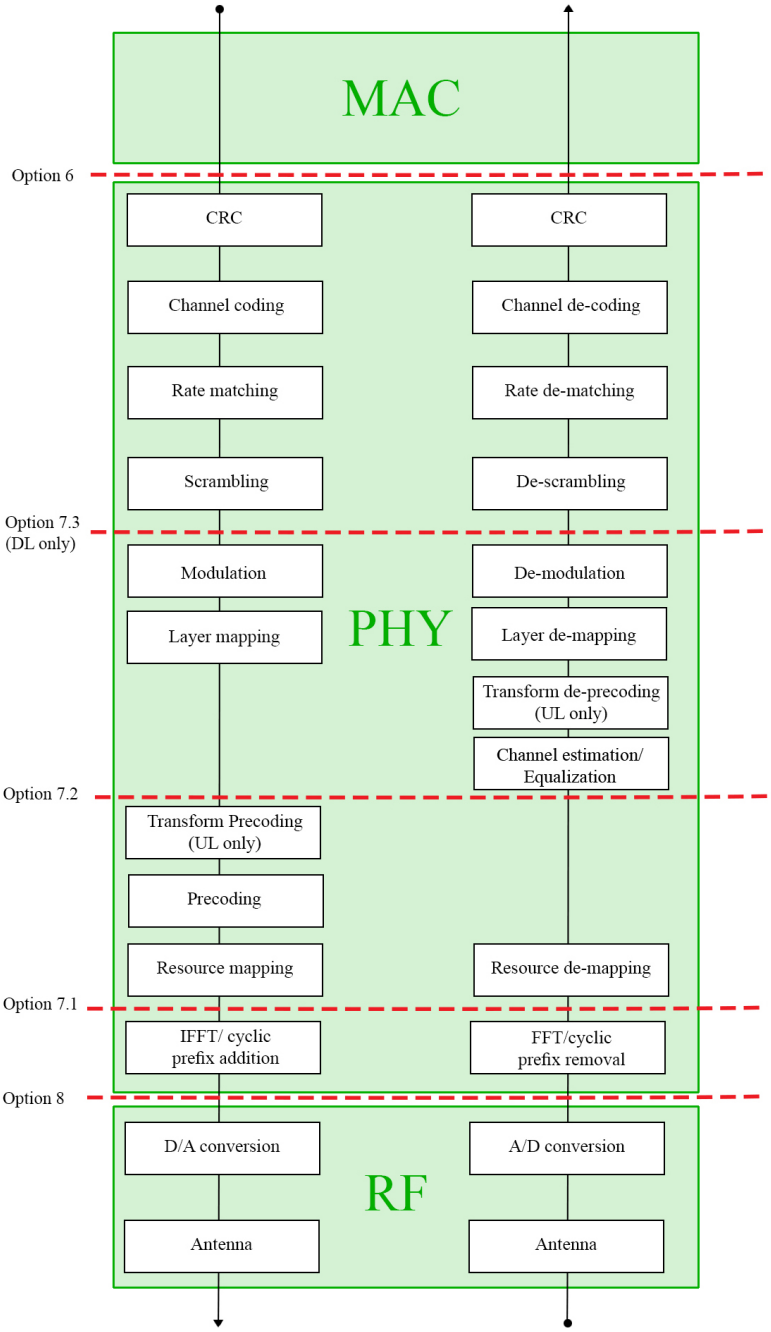

Figure 1.6: Functional splits at physical layer level. DL represents downlink, UL represents uplink (adapted from [7]).

1.3 Objectives

The main purpose of this dissertation is to implement a system of an uplink transmitter for UE, capable of processing LTE signals, following the technical specifications of *Third Generation Partnership Project* (3GPP). This purpose is divided into the following seven points:

- Understanding the base concepts, specifications, architectures and technologies for LTE networks, in particular the physical layer chain system of LTE for uplink communications;
- Define the system structure and signals to be transmitted;
- Design the system for an uplink transmitter in Simulink, integrated in MATLAB;
- Simulation and validation of the Simulink model;
- Generate VHDL code from the Simulink model through the HDL Coder tool in MATLAB;
- Simulation and validation of the VHDL code generated through a testbench;
- Discuss the results obtained from the two validated simulations.

1.4 Structure

Given the context and objectives for the subject of this dissertation, the remainder of this document is composed by four chapters. The structure for each chapter is defined as follows:

- The second chapter, named "LTE Uplink Subsystem", describes the basic notions about LTE technologies, more specifically the LTE specifications, the different types of information that are processed, how the information to be transmitted is organized and how the transmission is processed for uplink communications.
- The third chapter, named "System Modelling", shows the defined system structure for a LTE Uplink transmitter, proposed for the subject of this dissertation. It is explained the use of Simulink for its design, as well as presenting a detailed explanation of each component present in the proposed system.

- The fourth chapter, named "Results", describes the two simulation processes in Simulink and in VHDL testbench, showing the results' analysis from both simulations. It also shows resource utilization estimates from HDL Coder and VHDL synthesis.
- The fifth chapter, named "Conclusion", gives the conclusions about the developed work and possible suggestions for future works that can be developed, from the work of this dissertation.

Chapter 2

LTE Uplink Subsystem

2.1 Introduction

The objective of this chapter is to present the *standard* features and physical layer specifications of LTE, according to the 3GPP specifications, more specifically for uplink communications.

The topics addressed in this chapter are going to be LTE transmission frequencies on Frequency Division Duplex (FDD) mode and Time Division Duplex (TDD) mode, LTE bandwidths, frame structure, resource grid, uplink channels and reference signals, and the frequency-division multiple access scheme used in uplink communications.

The intended purpose of this chapter is to show the physical layer processing chain as well as to understand the contents of the developed work shown on chapter 3.

2.2 LTE Transmission Frequencies

The LTE transmission frequencies can be on different operating frequency bands, as defined by 3GPP *standards*. The operating frequency bands on FDD mode for uplink and downlink connections are separated, allowing for them to be transmitted simultaneously in time. On TDD mode, the uplink and downlink share the same operating frequency bands, which does not allow for simultaneous transmission of both. The 3GPP specifications, for LTE and LTE-Advanced operating bands, are shown in table A.1 and table A.2, in the Appendices section.

2.3 LTE Bandwidths

The bandwidth allocated to LTE communications is defined between 1.4 MHz and 20 MHz, according to 3GPP *standards*. The bandwidth is formed by smaller, contiguous frequency sections known as resource blocks. Each resource block is divided into 12 subcarriers, each subcarrier separated from each other by 15 KHz, which makes the subcarriers orthogonal to each other in the frequency domain. So, each resource block bandwidth is 180 KHz, allowing bandwidths with resource blocks ranging from 6 to 100.

For bandwidths between 3 MHz and 20 MHz, the resource blocks occupy 90% of the total bandwidth. For 1.4 MHz, the resource blocks occupy 77% of the bandwidth.

Table 2.1: Resource blocks in channel bandwidths.

Bandwidth (MHz)	**Resource blocks**	**Occupied bandwidth (%)**
1.4	6	77
3	15	90
5	25	90
10	50	90
15	75	90
20	100	90

2.4 Frame structure

Uplink transmissions are organized as radio frame sequences of 10 ms each. There are two types of frame structure: type 1 and type 2.

2.4.1 Frame structure type 1

The frame structure type 1 is applicable to both full duplex and half duplex FDD. In half-duplex FDD operation, the UE cannot transmit and receive at the same time while there are no such restrictions in full-duplex FDD [9]. Because it is applicable in FDD, uplink and downlink are separated in the frequency domain.

In each radio frame, there are 10 subframes, each with 1 ms of duration. A subframe is composed by two slots, during 0.5 ms each. Each slot is composed by 6 or 7 SC-FDMA symbols, for extended or normal cyclic prefix, respectively. Also, each symbol carries its own cyclic prefix with a certain length, which is explained in the section 2.9. The figure 2.1

presents the frame structure for type 1.

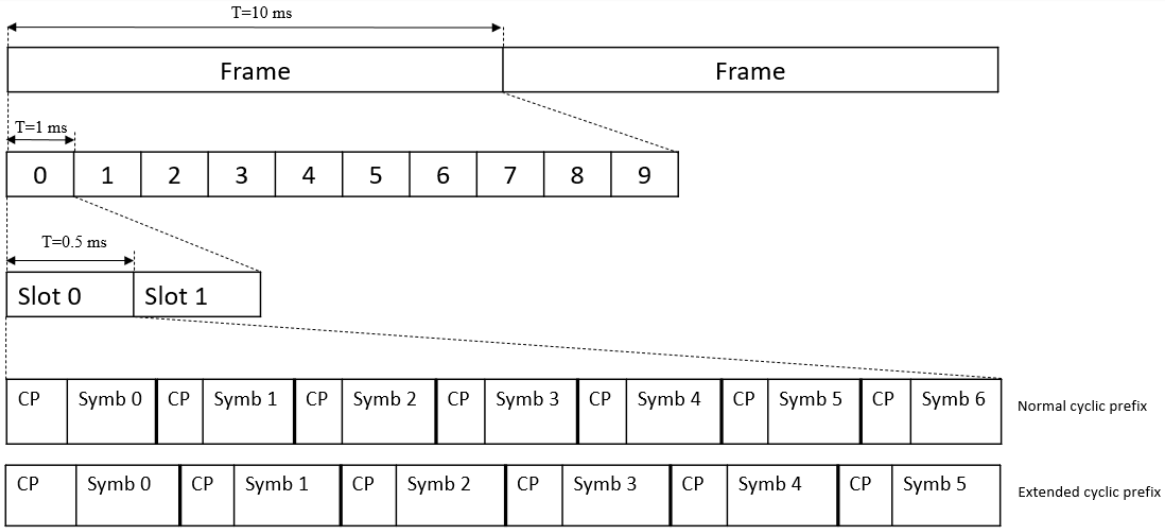

Figure 2.1: Frame structure type 1.

2.4.2 Frame structure type 2

The frame structure type 2 is used on TDD. Each frame is composed by 2 half-frames with duration of 5 ms each. Each half-frame is organized by 5 subframes, during 1 ms each. Finally, each subframe is composed by 6 or 7 symbols, for Extended or Normal Cyclic Prefix, respectively.

There are 3 kinds of subframe configuration: a subframe reserved for downlink transmission; a subframe reserved for uplink transmission and a special subframe with three fields: Download Pilot Time Slot (DwPTS), Guard Period (GP) and Uplink Pilot Time Slot (UpPTS). DwPTS contains reference signals, control information and synchronization signals for downlink. UpPTS is reserved for sounding reference signal (SRS) and Random Access Channel (RACH) transmission. GP defines the switching time between downlink and uplink transmission.

The table 2.2 shows the configuration of special subframe, with values in time domain expressed in time units of $T_s = \frac{1}{15000 \times 2048}$ s. The uplink and downlink configuration on each of the 10 subframes are shown in table 2.3, with 'D' representing a downlink subframe, 'U' representing an uplink subframe and 'S' representing a special subframe.

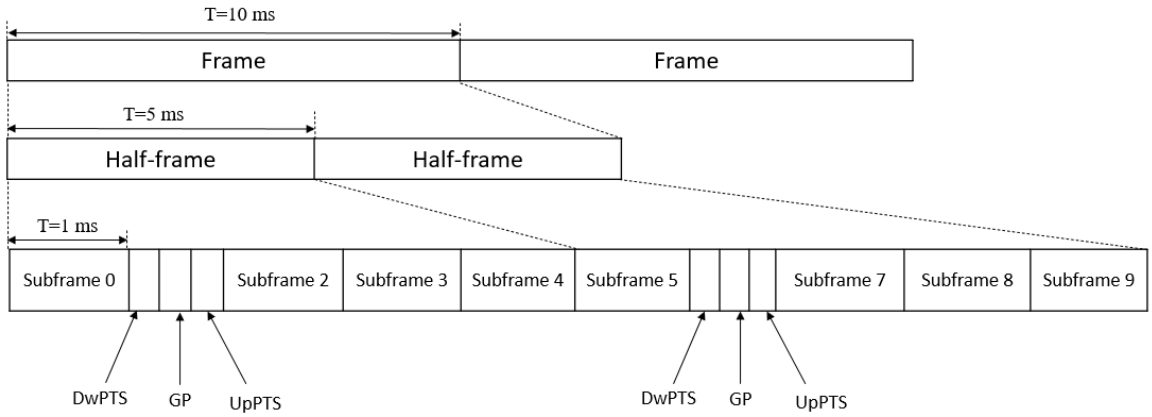

Figure 2.2: Frame structure type 2, for 5 ms switch-point periodicity (adapted from [9]).

Table 2.2: Configuration of special subframe (extracted from [9]).

Special subframe configuration	**Normal Cyclic Prefix in downlink**			**Extended Cyclic Prefix in downlink**		
	DwPTS	**UpPTS**		**DwPTS**	**UpPTS**	
		Normal Cyclic Prefix in Uplink	**Extended Cyclic Prefix in Uplink**		**Normal Cyclic Prefix in Uplink**	**Extended Cyclic Prefix in Uplink**
0	$6592.T_s$	$2192.T_s$	$2560.T_s$	$7680.T_s$	$2192.T_s$	$2560.T_s$
1	$19760.T_s$			$20480.T_s$		
2	$21952.T_s$			$23040.T_s$		
3	$24144.T_s$			$25600.T_s$		
4	$26336.T_s$			$7680.T_s$	$4384.T_s$	$5120.T_s$
5	$6592.T_s$	$4384.T_s$	$5120.T_s$	$20480.T_s$		
6	$19760.T_s$			$23040.T_s$		
7	$21952.T_s$			-	-	-
8	$24144.T_s$			-	-	-

Table 2.3: Uplink and downlink configurations (extracted from [9]).

Uplink-downlink configuration	Downlink to Uplink switch-point periodicity	Subframe number									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	5 ms	D	S	U	U	U	D	S	U	U	U
1	5 ms	D	S	U	U	D	D	S	U	U	D
2	5 ms	D	S	U	D	D	D	S	U	D	D
3	10 ms	D	S	U	U	U	D	D	D	D	D
4	10 ms	D	S	U	U	D	D	D	D	D	D
5	10 ms	D	S	U	D	D	D	D	D	D	D
6	5 ms	D	S	U	U	U	D	S	U	U	D

2.5 Resource Grid

In an LTE signal processing chain, it is important to organize the different resources into a structure that can be processed into a signal ready to be transmitted. This structure is known as a resource grid, where all the information coming from different channels and signals can be allocated in time and frequency.

The resource grid can be organized as the frame structure type 1, without the cyclic prefixes. Each SC-FDMA symbol in the resource grid extends over the avaliable bandwidth, ranging from 6 to 100 resource blocks, with each of the 12 subcarriers inside a resource block representing a resource element.

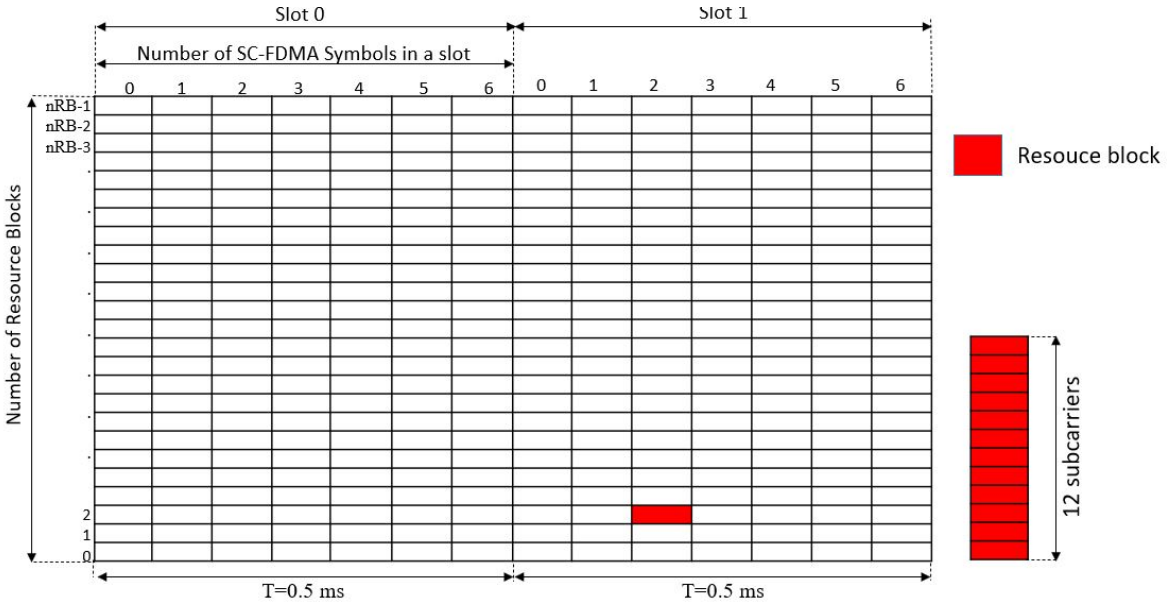

Figure 2.3: Resource grid structure, for normal cyclic prefix.

2.6 Uplink Channels

An uplink physical channel corresponds to a set of resource elements carrying information originating from higher layers [9]. The connections between physical channels and channels on higher layers are shown in figure 2.4.

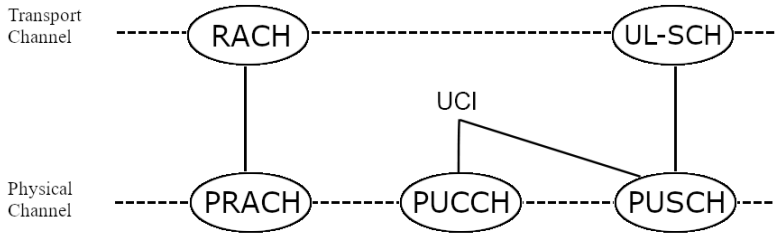

Figure 2.4: Uplink channel mapping.

2.6.1 Physical Uplink Shared Channel (PUSCH)

The Physical Uplink Shared Channel (PUSCH) carries Uplink Shared Channel (UL-SCH) data and can carry Uplink Control Information (UCI), depending on the UE configuration.

The allocation of PUSCH into the resource grid may take the whole or partial bandwidth available. PUSCH is allocated in all symbols in a slot, except for the symbol reserved for PUSCH-DMRS. The last symbol in a subframe may be reserved to the Sounding Reference Signal (SRS), if it is enabled by information from higher layers. If SRS is enabled, PUSCH cannot be allocated in the last symbol of a subframe. The figure 2.5 shows the resource grid with a possible allocation for PUSCH data.

The PUSCH processing system is presented in figure 2.6. The PUSCH symbol modulation can support QPSK, 16QAM and 64QAM. Layer mapping and Precoding process can be ignored, if the data is processed for a single antenna slot transmission.

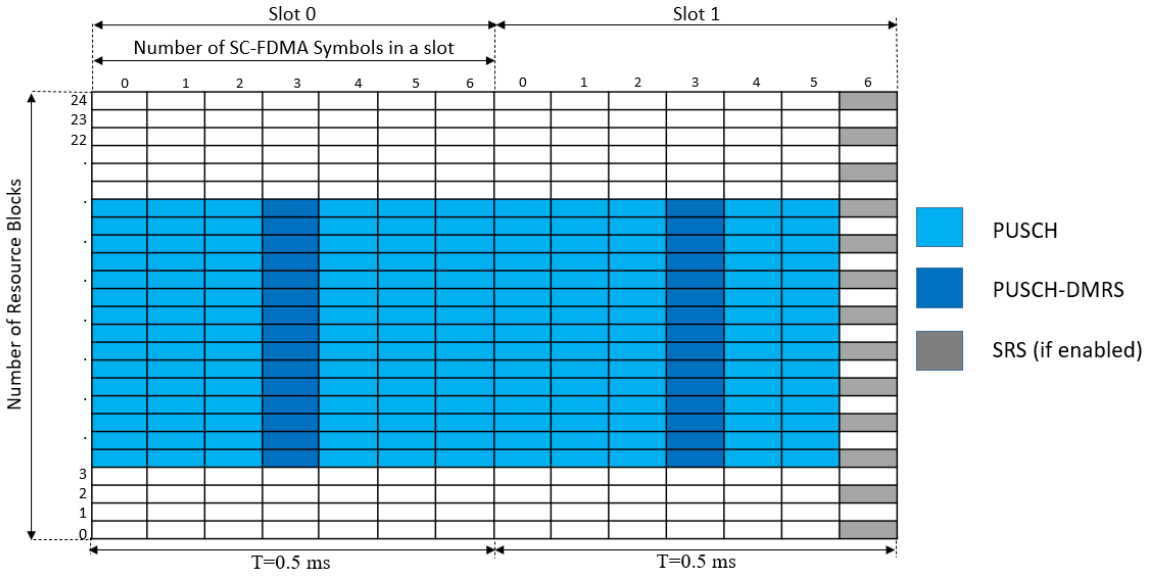

Figure 2.5: PUSCH in the resource grid, for bandwidth of 5 MHz, with Normal Cyclic Prefix.

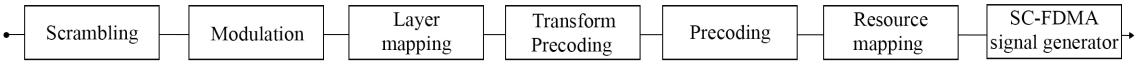

Figure 2.6: PUSCH processing system.

2.6.2 Physical Uplink Control Channel (PUCCH)

The Physical Uplink Control Channel (PUCCH) carries UCI. This channel supports different formats, depending on the information present in the UCI. The information each PUCCH format contains is presented in table 2.4, according to 3GPP specifications [10].

The allocation of PUCCH can be at one of the two resource blocks inside of a slot, located either at the upper or lower edge of the available bandwidth in the resource grid. On a subframe, if the one resource block used by PUCCH is located at the upper edge in one slot, the resource block for PUCCH on the other slot is at the lower edge. Simultaneous transmission of PUCCH and PUSCH is not allowed on LTE *standards*, but for LTE-Advanced *standards* it is possible, if enabled by higher layers. Like the PUSCH allocation, the last symbol in a subframe may be reserved for SRS, so that PUCCH cannot occupy the last symbol in a subframe. The location of different PUCCH formats with its PUCCH-DMRS, within a resource block, are shown in figure 2.8. The PUCCH format 2a and 2b only supports Normal Cyclic Prefix.

There are three processing systems for PUCCH formats, illustrated in figures 2.9, 2.10 and 2.11. All PUCCH formats, with the exception of PUCCH format 1, support BPSK and QPSK modulation.

Table 2.4: PUCCH format information.

PUCCH Format		**Number of bits**	**UCI Information**
Format 1		N/A	Only positive SR
Format 1a		1	1-bit HARQ-ACK or 1-bit HARQ-ACK with positive SR for FDD
Format 1b		2	2-bit HARQ-ACK or for 2-bit HARQ-ACK with positive SR
		4	4-bit HARQ-ACK with channel selection
Format 2		20	CSI report when not multiplexed with HARQ-ACK
		20	CSI report when multiplexed with HARQ-ACK for Extended Cyclic Prefix
Format 2a		21	CSI report multiplexed with 1-bit HARQ-ACK for Normal Cyclic Prefix
Format 2b		22	CSI report multiplexed with 2-bit HARQ-ACK for Normal Cyclic Prefix
Format 3	FDD	10	10-bit HARQ-ACK
		11	10-bit HARQ-ACK and 1-bit positive/negative SR
	TDD	20	20-bit HARQ-ACK
		21	20-bit HARQ-ACK and 1-bit positive/negative SR

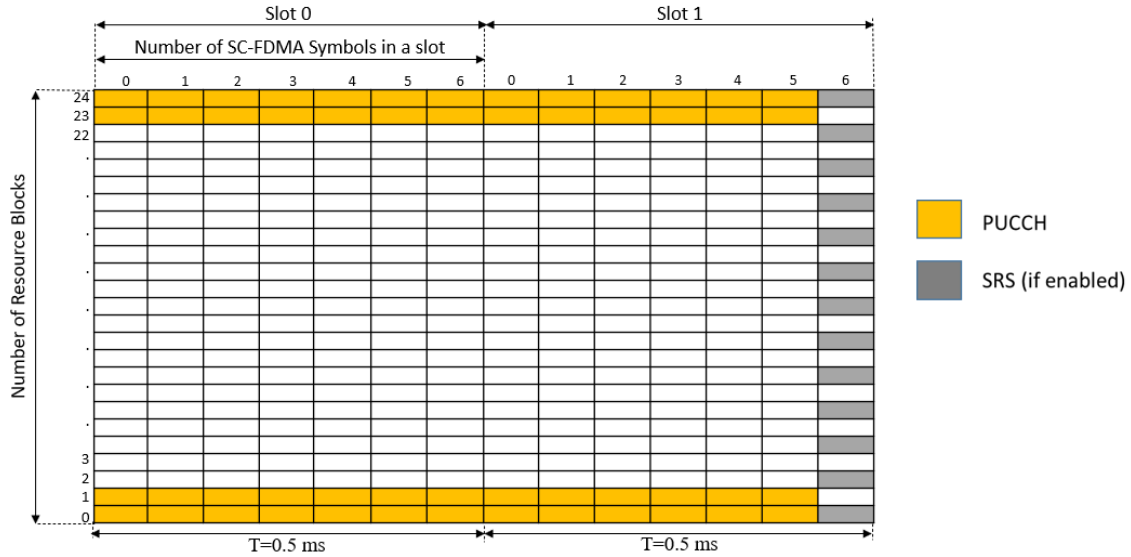

Figure 2.7: PUCCH in the resource grid, for bandwidth of 5 MHz, with Normal Cyclic Prefix.

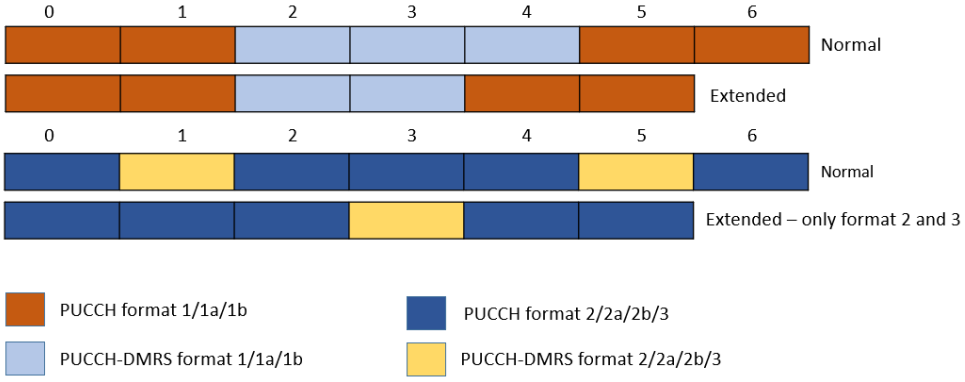

Figure 2.8: PUCCH formats in a resource block.

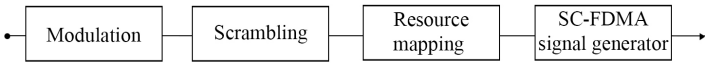

Figure 2.9: PUCCH format 1 processing system.

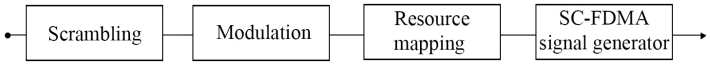

Figure 2.10: PUCCH format 2 processing system.

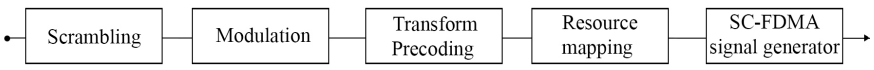

Figure 2.11: PUCCH format 3 processing system.

2.6.3 Physical Random Access Channel (PRACH)

The Physical Uplink Control Channel (PRACH) carries Randon Access Channel (RACH) preamble data, with the purpose to initiate connection between UE and base station and establish connection to the network. The preamble in PRACH is composed by a cyclic prefix and a sequence, as shown in figure 2.12. The length of cyclic prefix, defined as T_{cp}, and the length of the sequence, defined as T_{seq}, depends on the preamble format, as table 2.5 shows. Preamble format 4 is only applicable on frame structure type 2.

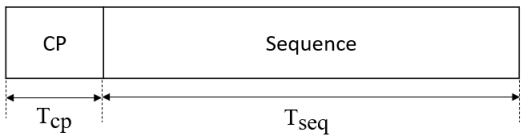

Figure 2.12: PRACH preamble format (extracted from [9]).

Table 2.5: Cyclic prefix and sequence lengths for PRACH preamble formats (extracted from [9]).

Preamble format	T_{cp}	T_{seq}
0	$3168.T_s$	$24576.T_s$
1	$21024.T_s$	$24576.T_s$
2	$6240.T_s$	$2.24576.T_s$
3	$21024.T_s$	$2.24576.T_s$
4	$448.T_s$	$4096.T_s$

The random access preambles are generated from Zadoff-Chu sequences with zero correlation zone, generated from one or several root Zadoff-Chu sequences [9]. The subcarrier spacing in PRACH preamble is 1250 Hz, for preamble formats 0 to 3, while the subcarrier spacing for preamble format 4 is 7500 Hz. For both frame structures, the preamble occupies the equivalent to 6 consecutive resource blocks. The positioning in the frequency domain of the PRACH preamble on the resource grid depends on the resource block offset, configured by higher layers, which corresponds to the first resource block available for PRACH. There are specific subframes for which PRACH is transmitted, depending on the PRACH configuration parameter, provided by higher layers.

The PRACH allocation on the resource grid in a subframe is displayed in figure 2.13, and the processing blocks for PRACH is shown in figure 2.14.

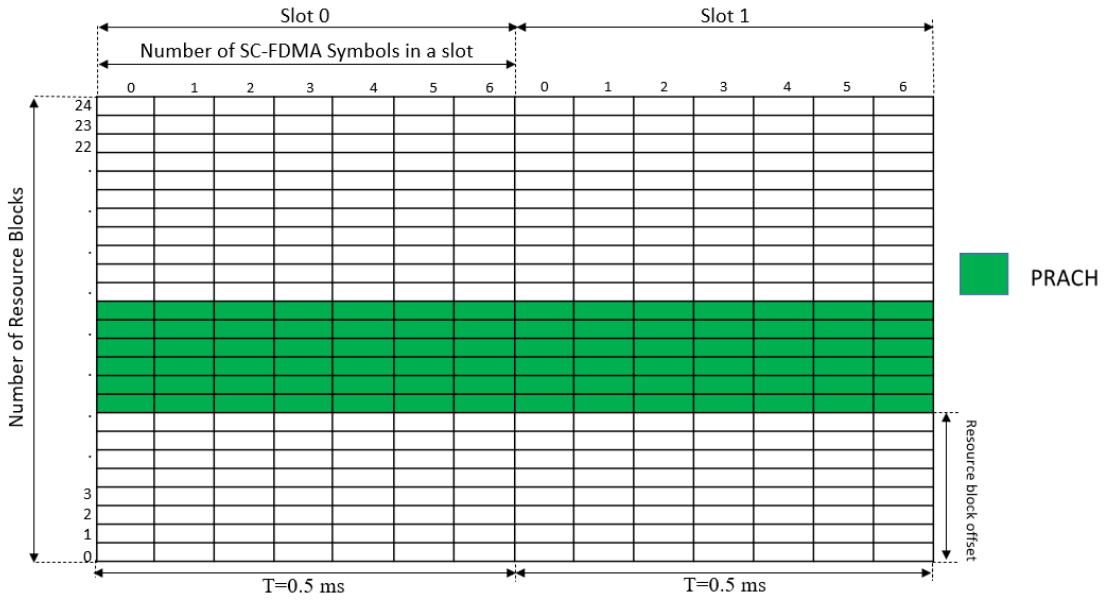

Figure 2.13: PRACH in the resource grid, for bandwidth of 5 MHz, with Normal Cyclic Prefix.

Figure 2.14: PRACH processing system.

2.7 Reference Signals

An uplink reference signal is used by the physical layer but does not carry information originating from higher layers [9]. There are two different reference signals:

- Demodulation Reference Signal (DMRS);
- Sounding Reference Signal (SRS).

2.7.1 Demodulation Reference Signal (DMRS)

The PUSCH and PUCCH are associated and transmitted simultaneously with their own DMRS and its intended use is to allow the base station, acting as a receiver, to decode properly the PUSCH and PUCCH information. The DMRS is a Zadoff-Chu base sequence with a cyclic shift specific for either PUSCH or PUCCH. The location of DMRS is illustrated

in figures 2.5 and 2.8, for PUSCH and PUCCH, respectively. The allocation of both DMRS in specific symbols on a slot is shown in table 2.6, for Normal and Extended Cyclic Prefix.

Table 2.6: DMRS symbol locations on a slot of a resource grid, for PUSCH and PUCCH.

	Normal Cyclic Prefix	**Extended Cyclic Prefix**
PUSCH	3	2
PUCCH format 1/1a/1b	2, 3, 4	2, 3
PUCCH format 2/3	1, 5	3
PUCCH format 2a/2b	1, 5	-

2.7.2 Sounding Reference Signal (SRS)

Unlike DMRS, the SRS is not associated with PUSCH or PUCCH. SRS is a reference signal transmitted by the UE, so the base station can estimate uplink channel quality at different frequency regions. The base station can then send information about the frequency region which suits best for the UE to transmit over. The SRS, if enabled by higher layers, is allocated on the last symbol of a subframe, as illustrated in figure 2.5 and 2.7. There are specific subframes for which SRS is transmitted, depending on the SRS subframe configuration parameter, configured by higher layers. The figure 2.15 shows an example of a SRS configuration, for SRS allocation on the first two subframes only, for frame structure type 1.

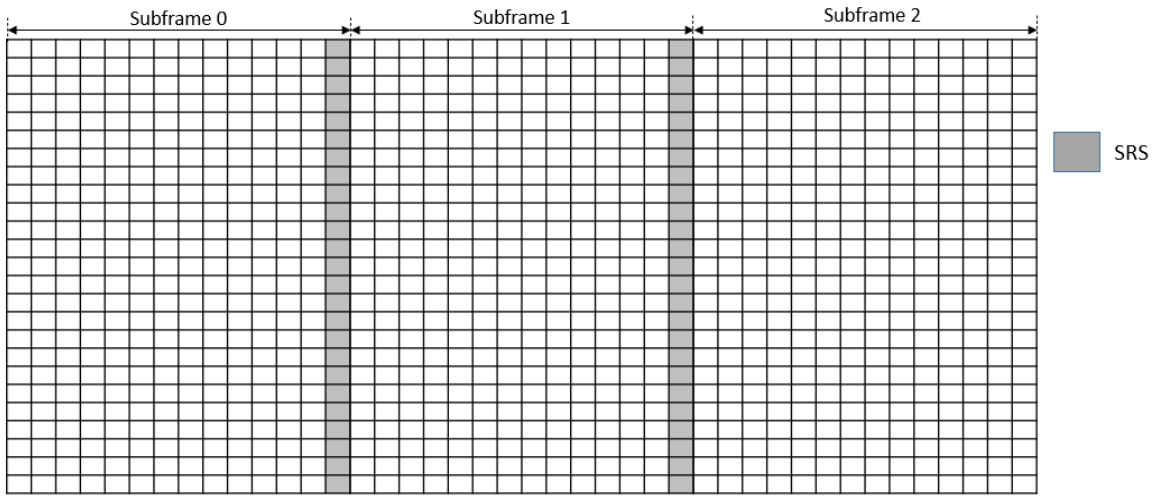

Figure 2.15: SRS allocation on the first two subframes, for frame structure type 1.

2.8 Single-carrier Frequency-Division Multiple Access (SC-FDMA)

The Single-carrier Frequency-Division Multiple Access (SC-FDMA) is the variant scheme of OFDMA, used for uplink transmissions. The advantage of using SC-FDMA over OFDMA is the lower Peak-to-Average Power Ratio (PAPR) than OFDMA scheme, reducing the linearity on the power amplifier design at the RF chain, increasing the power efficiency and reducing the power consumption on devices with limited power consumption. SC-FDMA scheme follows the OFDMA scheme, with the addition of DFT precoding, before subcarrier mapping into the resource grid, which is known as the Transform Precoding processing function on the uplink transmitter system. By applying a DFT-based precoding, SC-FDMA substantially reduces fluctuations of the transmit power, transforming data in time domain into frequency domain at the DFT output and creating a single carrier signal. The resulting uplink transmission scheme can still feature most of the benefits associated with OFDMA, such as low-complexity frequency-domain equalization and frequency-domain scheduling, with less stringent requirements on the power amplifier design [1].

Like OFDM, SC-FDMA divides the transmission bandwidth into multiple parallel subcarriers, with the orthogonality between the subcarriers being maintained in frequency-selective channels by the use of a cyclic prefix or guard period [11].

In the SC-FDMA, there are two types of subcarrier mapping: localized and distributed mapping. In localized mapping, a group of adjacent subcarriers are allocated to a user, while in distributed mapping the subcarriers allocatted are equally spaced between each other. The figure 2.16 shows the SC-FDMA processing system, with both subcarrier mapping schemes.

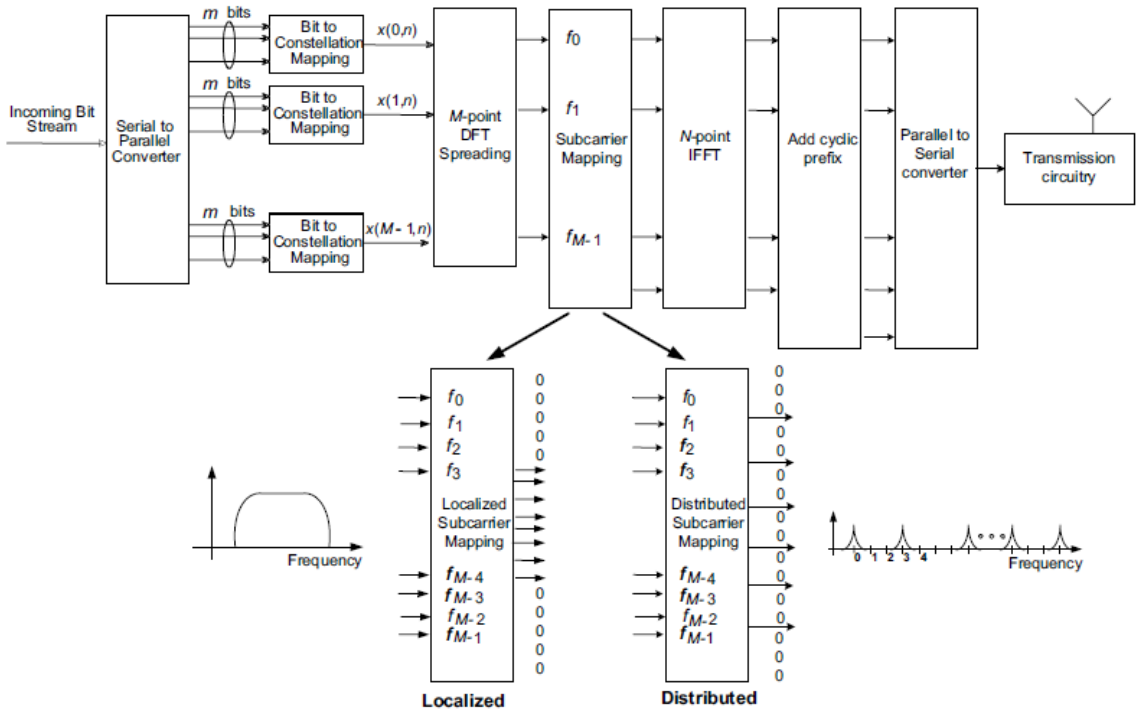

Figure 2.16: SC-FDMA processing system, showing localized and distributed subcarrier mappings (extracted from [11]).

2.9 Cyclic Prefix

The cyclic prefix is the data insertion of a certain last number of samples of a symbol, before the initial sample of the same symbol. The use of a cyclic prefix prevents Inter-Symbol Interference (ISI) between SC-FDMA information blocks. It transforms the linear convolution of the multipath channel into a circular convolution, enabling the receiver to equalize the channel simply by scaling each subcarrier by a complex gain factor [11].

The frame structure type 1 section mentioned two types of cyclic prefix: Normal Cyclic Prefix and Extended Cyclic Prefix. While Extended Cyclic Prefix has a larger data length than Normal Cyclic Prefix, the length of one slot cannot be changed, so the number of symbols in a slot is decreased to compensate for the Extended Cyclic Prefix data length.

Inside the slot in a radio subframe, there are 7 symbols and 7 cyclic prefixes. For Normal Cyclic Prefix, the first cyclic prefix, corresponding to the first symbol in the sequence, has a longer length than the remaining cyclic prefixes. The sample length varies with IFFT bin size for each SC-FDMA symbol. The time and sample length for each cyclic prefix inside a time

slot, for Normal and Extended Cyclic Prefix, are shown in table 2.7, with associated uplink parameters.

Table 2.7: Cyclic prefix lengths, with uplink parameters. The l stands for symbol index.

Bandwidth		1.4 MHz	3 MHz	5 MHz	10 MHz	15 MHz	20 MHz
Allocated resource blocks		6	15	25	50	75	100
Number of subcarriers/symbol		72	180	300	600	900	1200
IFFT size		128	256	512	1024	1024	2048
Sampling frequency		1.92 MHz	3.84 MHz	7.68 MHz	15.36 MHz	15.36 MHz	30.72 MHz
Normal Cyclic Prefix	**time**	5.2 us, for $l = 0$; 4.7 us for l = 1:6					
	samples	10 for $l = 0$ 9 for l = 1:6	20 for $l = 0$ 18 for l = 1:6	40 for $l = 0$ 36 for l = 1:6	80 for $l = 0$ 72 for l = 1:6	80 for $l = 0$ 72 for l = 1:6	160 for $l = 0$ 144 for l = 1:6
Extended Cyclic Prefix	**time**	16.7 us					
	samples	32	64	128	256	256	512

2.10 Related works

Inside the scope of LTE uplink transmitters, there are implementations proposed on several documents. This section presents some of the documents related to the work of this dissertation.

In [12] it is presented an uplink baseband signal generation model for the LTE UE transmitter, supporting the three physical uplink channels (PUSCH, PUCCH and PRACH), the Transform Precoding algorithm and hardware architectures for some of the blocks that compose the uplink transmitter system.

In [13] it is discussed a work developed in the scope of a master thesis, containing an overview of LTE concepts, complexity analysis of the algorithms involved in the LTE downlink and uplink subsystems, and a discussion on the hardware requirements based on the results of the analysis presented.

The master thesis [14] shows a SC-FDMA transmiter implementation using Software-Defined Radio (SDR). It presents SDR concepts and modules for OFDM (Orthogonal Frequency Division Multiplexing) and SC-FDMA transmission techniques. It also shows the OFDM and SC-FDMA system implementation and a comparative performance analysis.

All the related works cited in this section present methods to design and implement an uplink transmitter subsystem based on SC-FDMA techniques, supporting the three physical uplink channels. These implementations were the basis for the LTE uplink transmitter subsystem developed in the scope of this work. This dissertation presents a design of an uplink

transmitter system, detailing the required blocks, according to LTE *standards*. Additionally, the VHDL code generation blocks for hardware implementation are also discussed. The development of this model is fundamental for behavioral and cycle accurate system validation. It can also be used to estimate the hardware resources needed for its implementation, providing an essential baseline for future developments related to hardware implementation of an uplink transmitter, including in 4G/5G coexistence scenarios.

With the discussion of LTE Uplink subsystem features and physical layer specifications, the modellation of a LTE Uplink transmitter processing chain is then presented on the next chapter.

Chapter 3

System Modelling

3.1 Introduction

As discussed in the Chapter 1, about the processing chains possible of being implemented in hardware for an uplink transmitter, the SC-FDMA scheme defines the processing blocks for the design of an uplink transmitter, specifically for UE devices. For this reason, in the context of the dissertation, the design of an uplink transmitter system was developed, following the SC-FDMA scheme.

The design of each processing block for the system takes into account their response to input, meaning that during the design of each process that compose the system, their expected data output needs to validate the obtained data at their output. In order to optimize resources needed for an FPGA implementation, the system is designed to have bit widths as small as possible, for all signals. Because of this, the produced outputs by the system are not exactly equal to the expected outputs. The transmitter system is bound to produce precision errors, but that does not mean that the information in the transmitted data is compromised, when picked up by the receiver. As long as the receiver gets the intended data that was transmitted, the system design for the transmitter is not compromised.

In order to design a functional uplink transmitter, it needs a software with modelling, simulating and analyzing capabilities. It also needs software capable of automatic code generation, specifically for VHDL generation. Finally, it is needed comparable uplink data that is proven it is picked up properly by an uplink receiver. To fulfill this requirements, the MATLAB software was chosen. MATLAB has a tool for modelling, simulating and analyzing

data, which is Simulink. MATLAB has automatic VHDL code generator, which is the HDL coder. MATLAB also has waveform generator for uplink data, specifically LTE modulated waveform data, to compare its data against the data output at the designed system. This waveform generator viability for uplink data generation is discussed in chapter 4.

This chapter discusses the tools used within MATLAB, which are Simulink and HDL Coder, as well as to show the developed system with digital signal processing blocks and explain how these processing blocks work inside the system.

3.2 Simulink

Simulink is a graphic programming environment inside the MATLAB software, that allows for modelling, simulating and analyzing multidomain dynamic systems and embedded systems. It also has support for automatic code generation, continuous testing and verification. Being integrated in MATLAB, it is possible to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis [15]. Simulink also has customizable block libraries.

As with MATLAB, Simulink can handle various data types, signal dimensions and values. These data types can be single or double precision; signed or unsigned 8, 16 or 32 bit integers; boolean and fixed-point. Signal dimensions can be scalar, vector, matrix or multidimensional arrays. Values can be real or complex-valued.

In the context of the dissertation work, Simulink is used to develop the uplink transmitter system, since it has support for automatic code generation for VHDL, has HDL-compatible blocks for synthesizable VHDL and support for fixed-point data type, which is useful for processing data with bit widths as small as possible.

3.3 HDL Coder

The HDL Coder is a tool inside the MATLAB software that allows for automatic code generation of synthesizable VHDL and Verilog code from MATLAB functions, Simulink models and Stateflow charts. The generated HDL code can be used for FPGA programming or ASIC prototyping and design. HDL Coder provides a workflow advisor that automates the programming of Xilinx, Microsemi, and Intel FPGAs. It is possible to control HDL architecture and implementation, highlight critical paths, and generate hardware resource utilization

estimates. HDL Coder provides traceability between your Simulink model and the generated Verilog and VHDL code, enabling code verification for high-integrity applications adhering to DO-254 and other standards [16].

As stated before, Simulink has customizable block libraries. Some of its libraries are designed specifically for synthesizable VHDL and Verilog with HDL Coder. The use of such libraries for the system design allows the use of its own processing blocks without having the need to design one with its purpose, while also implementing these processing blocks without the worry of verifying if the HDL coder can synthesize these blocks. The HDL Coder has a HDL Workflow Advisor, which shows the steps followed by the HDL Coder to synthesize code and allows for setting up code generation options, as illustrated in figure 3.1.

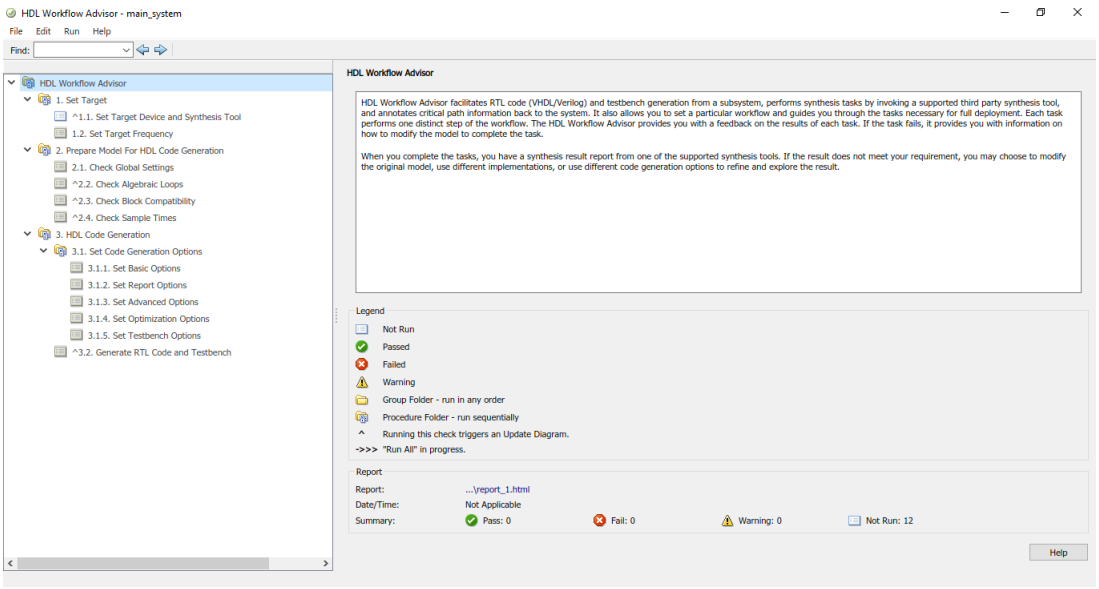

Figure 3.1: HDL Workflow Advisor on HDL Coder.

3.4 Physical Layer Uplink System

The system design is influenced by comparing uplink data generated with the data from the Waveform generator. Since the Waveform generator for uplink only generates PUSCH and PUSCH-DMRS data, it was decided to design the system for PUSCH information only. The system is also designed for single antenna port transmission, discarding the use of the Precoding process between Transform Precoding and Resource mapping process, as the Precoding process is used for spatial multiplexing to multiple antenna ports.

The physical layer uplink system, shown in figure 3.2, is responsible for processing PUSCH data, coming from the data output at the symbol-modulation process, generate PUSCH-DMRS data and output SC-FDMA waveform digital signal. Externally, this signal passes through the fronthaul interface, for it to be processed by the RF chain system.

The system design has processing functions which are modified versions of blocks, taken from the *LTE HDL PBCH Transmitter* system [17], a prebuilt LTE system for downlink data. The blocks that were adapted for the uplink system are the *Read Write Logic* and *LTE OFDM Modulator*.

Inside the physical layer uplink system designed for PUSCH data, the following blocks are present:

- *Transform Precoding:* performs amplitude scaling of modulated PUSCH complex data at the input, followed by the Fast Fourier Transform (FFT) processing; controls the data rate from the FFT output and also provides PUSCH allocation indices.
- *PUSCH-DMRS Generator:* generates the reference signal PUSCH-DMRS.
- *Resource Grid:* modified version of *Read Write Logic* used in *LTE HDL PBCH Transmitter* [17]. Writes PUSCH and PUSCH-DMRS data in specific addresses on the RAM block and reads the RAM block data, if it receives data request from the *SCFDMA Waveform Modulator* block, producing a resource grid symbol data at the output. It erases the RAM memory after reading a subframe of data.
- *SC-FDMA Waveform Modulator:* modified version of *LTE OFDM Modulator* used in *LTE HDL PBCH Transmitter* [17]. It performs SC-FDMA symbol formation, performs IFFT, shifts data by half the subcarrier spacing and in time domain consecutively, makes CP insertion, performs windowing and controls the data rate. The output is a SC-FDMA modulated signal.

The input of PUSCH data is complex-valued data, with fixed-point as its data type. Each sample of the PUSCH data is a signed 16-bit fixed-point value, with 1 bit for sign, 2 bits for integer representation and 13 bits for fractional representation. The data output also has the same number representation as the PUSCH data.

For the remainder of this dissertation, the physical layer uplink system designed is referred to as Uplink-Tx system.

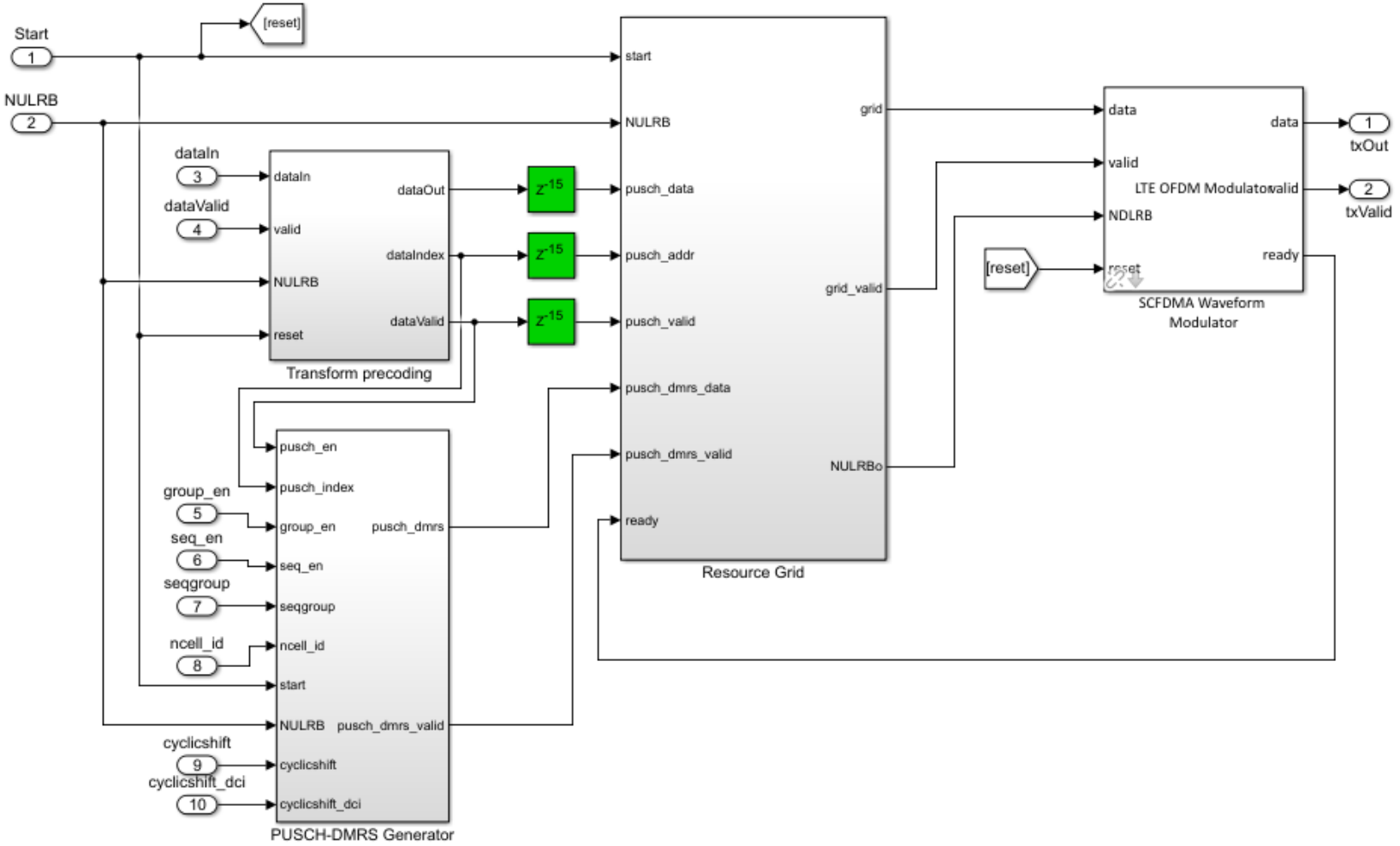

Figure 3.2: Uplink-Tx system with SC-FDMA processing, with support for PUSCH information only.

3.4.1 Transform Precoding

The *Transform Precoding* block, shown in figure 3.3, processes 12 PUSCH complex-data symbols, each with either 1200 or 1800 samples, before being stored in the *Resource grid* block.

The complex input data is first processed into the *Amplitude Scaling*, reducing the real and imaginary component with a multiplication by a factor of $\frac{1}{\sqrt{M_{sc}^{PUSCH}}}$, with M_{sc}^{PUSCH} representing the number of subcarriers for PUSCH data in a symbol. There are preserved values of this multiplying factor on a Lookup Table, mapped to the number of resource blocks allocated for PUSCH, at the input `NURLB`, preventing the use of a square root operation, followed by a division operation. The *Amplitude Scaling* block precedes the *1200-1800 point FFT* block, to prevent saturation on overflow data. However, the data at the output of *1200-1800 point FFT* system would still have saturation on overflow data. So, a shift aritmetric right operation was implemented inside the *Amplitude Scaling* block, to remove such overflow.

The *1200-1800 point FFT* performs FFT on data with size of 1200 or 1800 samples, depending on the resource blocks allocated for PUSCH. It either performs 1200-point FFT, for 25, 50 or 100 resource block allocation for PUSCH, or performs 1800-point FFT, for 6, 15 or 75 resource block allocation for PUSCH. If the allocation size is less than 100 resource blocks, the PUSCH data needs to be upsampled, by having zeros inserted between samples, before it enters the Uplink-Tx system. More details about the *1200-1800 point FFT* process is explained in section 3.4.2.

The *Base Rate Controller* is adapted from the *BaseRateController* in the *SCFDMA Waveform Modulator*. It receives either 1200 or 1800 consecutive samples from the *1200-1800 point FFT* block and downsamples it to either 72, 180, 300, 600 or 900 samples, reducing the number of valid samples at the input passing to the output. The number of valid samples at the output is equal to the 12*number of resource blocks. If the number of resource blocks are 100, it passes all the input to the output. The output then goes through the shift aritmetric left block, to cancel out the shift aritmetric right operation done at the *Amplitude Scaling* block, then it goes to the `dataOut` output.

The *PUSCH Index Generator* generates PUSCH resource element indices, given the number of resource blocks, and are synchronized with the data samples going to the `dataOut` output, with the introduction of one sample delay.

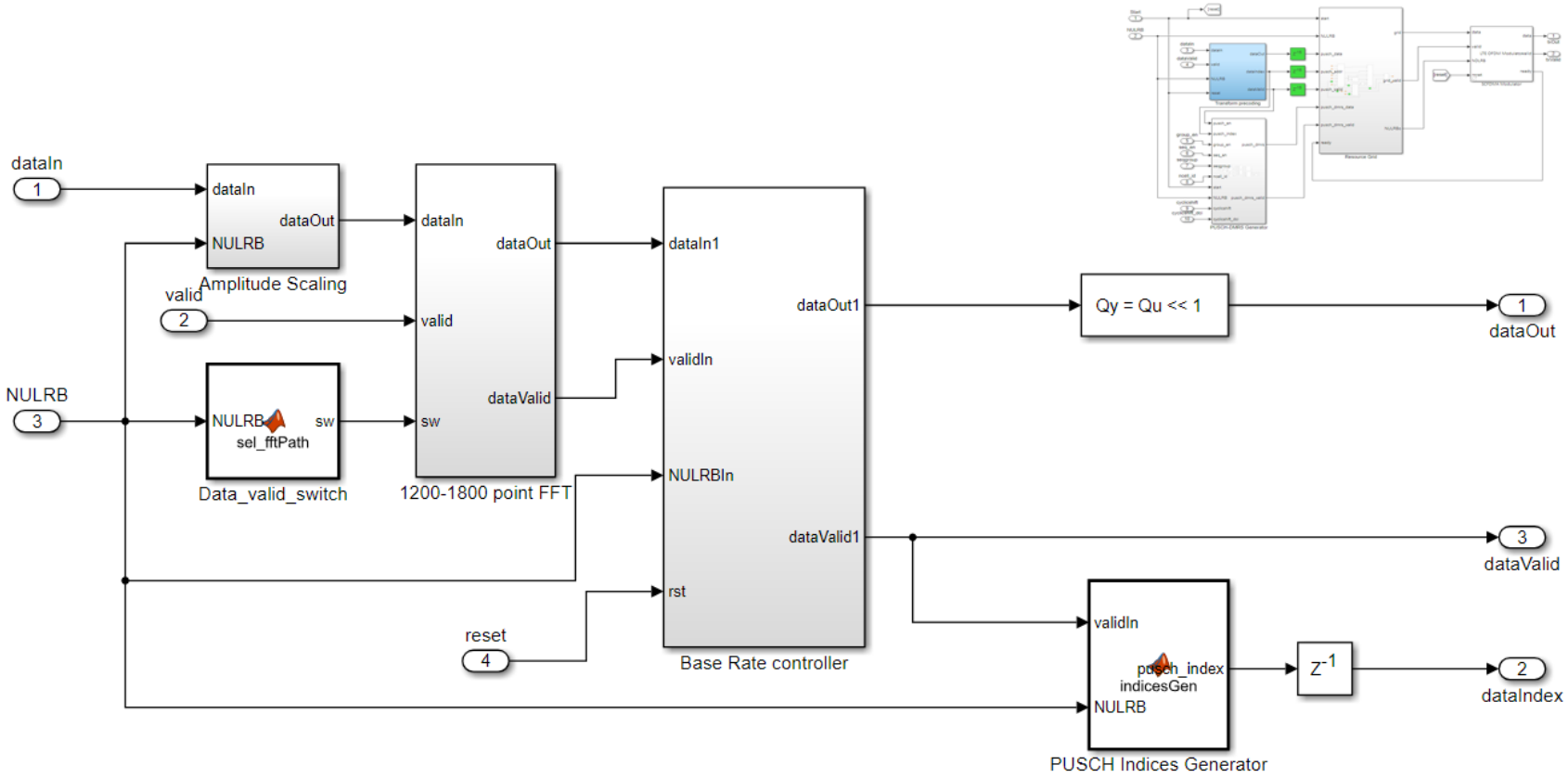

Figure 3.3: Transform Precoding block.

3.4.2 *1200-1800 point FFT*

The use of 1200 or 1800-point FFT covers 6 different sizes for data allocation in a PUSCH symbol: 72, 180, 300, 600, 900 and 1200. The FFT design follows the Cooley-Tukey method, as shown in figure 3.4. The FFT bin of size N is divided into two smaller FFT bins of sizes N1 and N2, so that $N = N1 \times N2$, meaning the product of the sizes of the smaller FFTs equals the size of the original FFT. The FFT bin size can be break down into prime factors of the original FFT. The 1200-point FFT size can be broken down into the product of $2^4 \times 3^1 \times 5^2$, while the 1800-point FFT size can be broken down into the product of $2^3 \times 3^2 \times 5^2$. Hence, it is needed 7 FFT stages, with the first FFT stage being either 2-point FFT or 3-point FFT. More details about 2, 3 and 5-point FFT operation can be consulted on [18].

As seen in figure 3.5, there is two function blocks, to select the stage where the PUSCH signal goes first, between a 2-point FFT or 3-point FFT, and to select the data processed from one of those stages. The *Stage 2-7 (600 Point FFT)* is a series of FFT stage processing blocks that perform 600-point FFT. The *Data Ordering* block is the equivalent to matrix transpose, after performing the FFT for each stage.

Each FFT stage is composed by a *radix-n FFT* block and a *twiddle factor n* block, as shown in figure 3.6. Inside the *radix-n FFT* block, the `dataIn` signal is synchronized with the `valid` signal. This `valid` signal, being of boolean type, activates an incremental counter, which tells the *Write/Read Control* block the index of the sample given the number of `valid` samples that are signaled *high*. The *Delay Line* stores data inside FIFO memory blocks, until it reaches its maximum limit, as long as the *Write/Read Control* gives the order to store incoming data. When the FIFO memory limit is reached, the *Write/Read Control* block sends a signal to pop out all the values stored in the FIFO memory. As stored data is popped out, the *n-point FFT* then performs its FFT operation of each of the data samples from the dataIn signal and the FIFO data samples. This whole operation performs the equivalent to step 2 and 3 in figure 3.4. Following this operation, the data goes through the *twiddle factor n* block, which is multiplying the data by a stored complex number, performing twiddle factor multiplication. Multiplying two complex numbers would normally take 4 multiplications and 2 additions, but this block performs 3 multiplications and 3 additions, so the number of multiplications are reduced. As the Cooley-Tukey algorithm is recursive, the step 5 in figure 3.4 can be seen as a series of FFTs going through step 1 to 4.

Inside the *Data reordering* block, shown in figure 3.7, there is a series of *Data reordering from stage n* blocks for each FFT stage performed, with the exception of the first block. Each *Data reordering from stage n* block has a RAM memory with write/read data in specific addresses that processes the signal as it makes a matrix transpose of the stored data. The *RAM controller* block is responsible for cleaning the RAM memories inside the data reordering blocks, after the *1200-1800 point FFT* has finished processing 12 symbols reserved for the PUSCH data samples. This ensures that the RAM does not keep stored data, if there is no more data to process until the next 12 symbols arrives at the block.

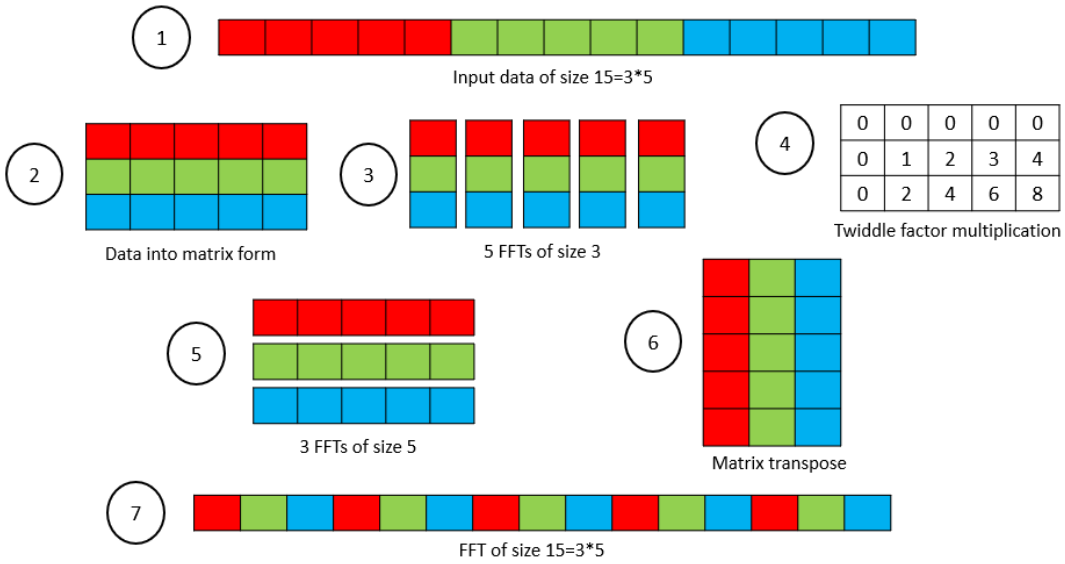

Figure 3.4: Cooley-Tukey FFT algorithm, exemplifying a 15-point FFT (adapted from [19]).

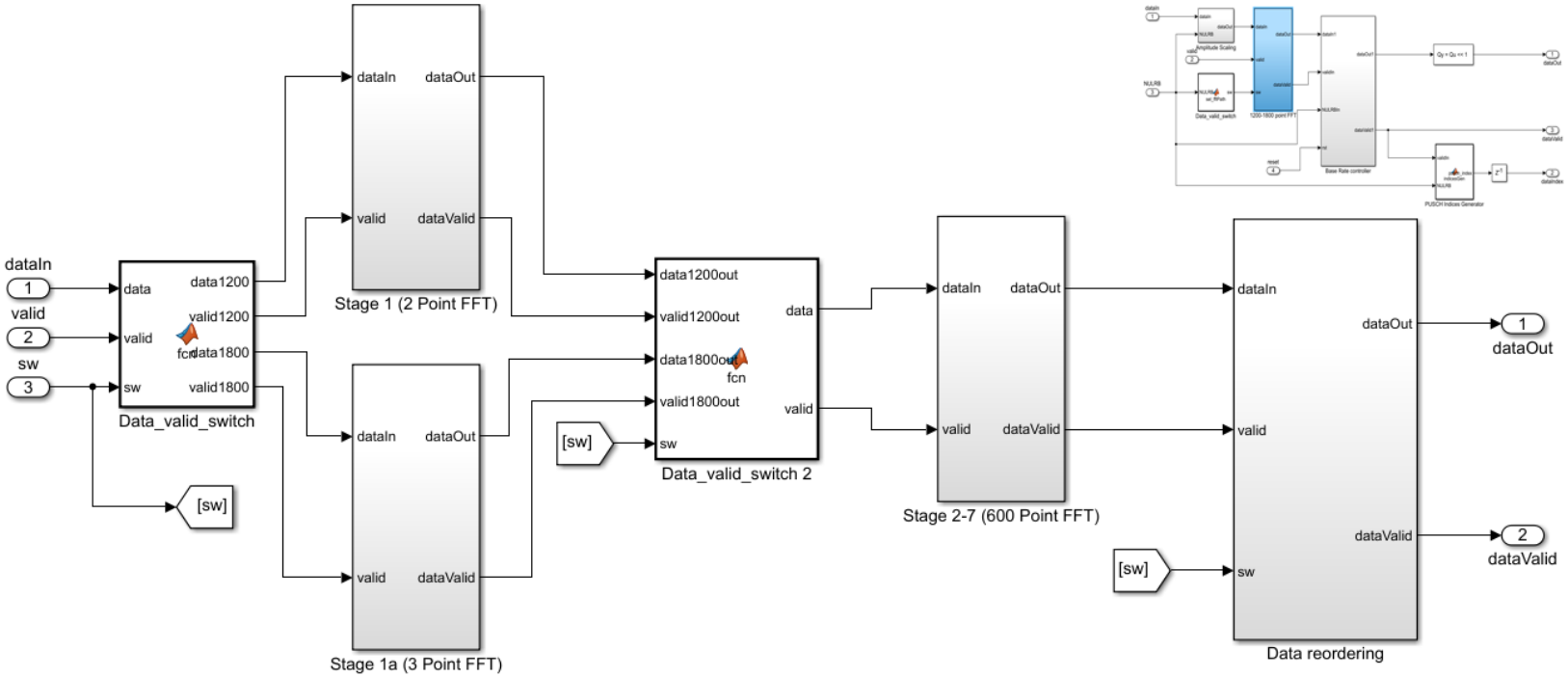

Figure 3.5: 1200-1800 point FFT block.

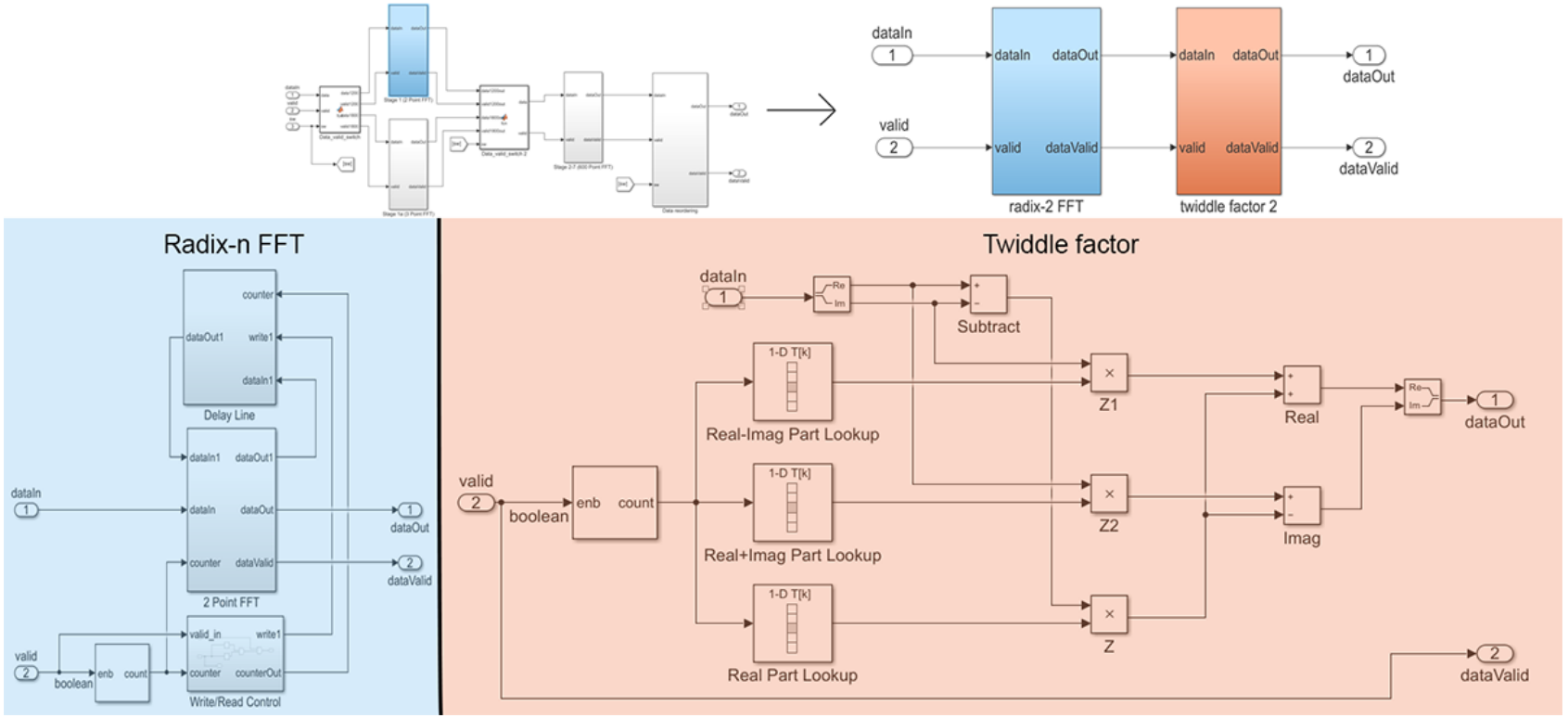

Figure 3.6: FFT stage processing block.

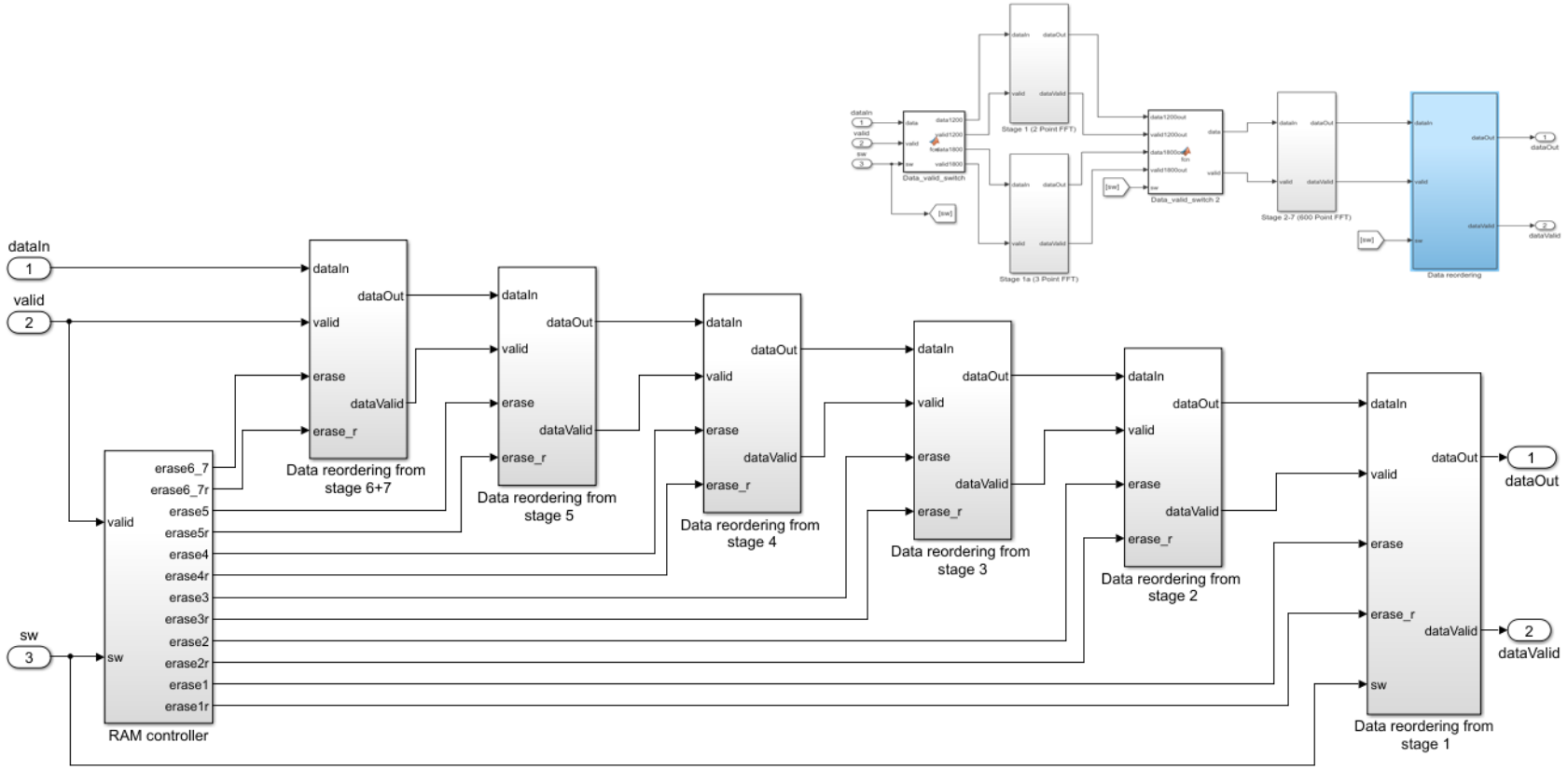

Figure 3.7: Data ordering block.

3.4.3 PUSCH-DMRS generator

The *PUSCH-DMRS generator*, as shown in figure 3.8, is a Zadoff-Chu sequence generator for PUSCH. Its operation is active after 2 PUSCH symbols are picked up by the Uplink-Tx system, for each subframe processed. A series of 15 sample delays are implemented at the PUSCH data, PUSCH indices and PUSCH valid signal, between the *Transform Precoding* block and the *Resource grid* block, to compensate for the PUSCH-DMRS processing delay. The processing of the *PUSCH-DMRS generator* is complex, as it takes into account important parameters to determine the generation of Zadoff-Chu sequence. These important parameters are: Physical layer cell identity, labeled as `ncell_id`; PUSCH sequence group assignment, labeled as `seqgroup`; Number of cyclic shifts used for PUSCH DM-RS, in normal and DCI format, labeled as `cyclicshift` and `cyclicshift_dci`, respectively; Group hopping and sequence hopping *flag*, labeled as `group_en` and `seq_en`, respectively. More detailed information about the PUSCH-DMRS data processing, based on this parameters, can be seen in Section 5.5 on [9].

At first, the *Gold Sequence Inputs* block generates the initial sequence value, loading and enable values to the *Gold Sequence Generator*, denoted by `init`, `load` and `enable`, respectively. The *Subcarrier count* block generate address values, corresponding to the subcarrier placing for each PUSCH-DMRS sample, while the *NRSZC Subcarrier count* blocks generate address values between 0 and the largest prime number that is smaller than the maximum value of subcarriers for PUSCH. After the values from the *Gold Sequence Generator* and address values from *Subcarrier count* block are received, the *Hopping Numbers* block determines the group hopping value and sequence number, denoted by `u_out` and `v_out`, respectively.

The *alpha exponent* and *Xq(m)* blocks both generate their own exponent factor values and then both add up, as both blocks represent the multiplication of two exponential numbers. This sum goes through the Coordinate Rotation Digital Computer (CORDIC) processing block, which uses shift-add operations to produce an approximate result of the desired function. The function of this CORDIC is to calculate the complex exponential, with the input as the angle, and outputs data in the form of cos(angle)+j*sin(angle).

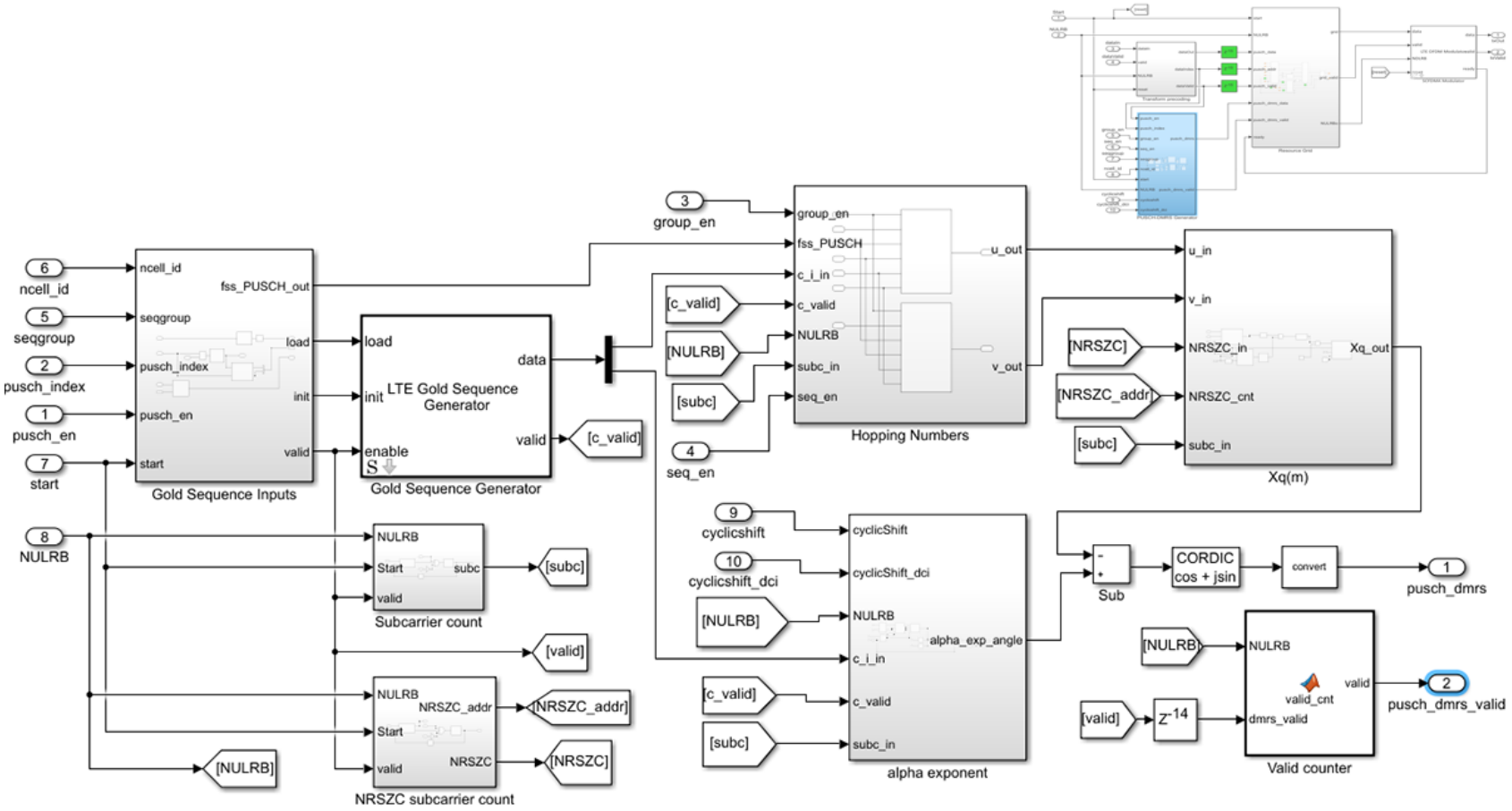

Figure 3.8: PUSCH-DMRS generator block.

3.4.4 Resource Grid

The *Resource Grid* block, as shown in figure 3.9, is responsible for storing a subframe of data into the RAM memory and read each of the symbols present in the subframe for the *SC-FDMA Waveform Modulator* to process.

The *Write Selector* writes subframes of data into the *LTE Memory Bank*. It first writes the first two symbols of PUSCH data and the two symbols of PUSCH-DMRS data simultaneously, then writes the remaining ten symbols of PUSCH data, sequentially. The PUSCH and PUSCH-DMRS data samples are written in memory banks, corresponding to the symbol number in a subframe, and written in the address inside of the memory bank, corresponding to the subcarrier index inside the symbol. The *LTE Memory Bank* can store up to 14*2048*16-bit complex values, meaning it can store 14 SC-FDMA symbols, each with 2048 complex values.

The *Read Selector* reads each of the symbols stored in the *LTE Memory Bank*, based on the `rd_enb` and `ready` signal from the *LTE SC-FDMA Modulator*. The *LTE Memory Bank* returns the data from all 14 memory banks. The *Grid Bank Select* chooses the appropriate symbol from these 14 banks to go to the output. After storing and reading a subframe of data, the *LTE Memory Bank* is erased, before writing the next subframe of data.

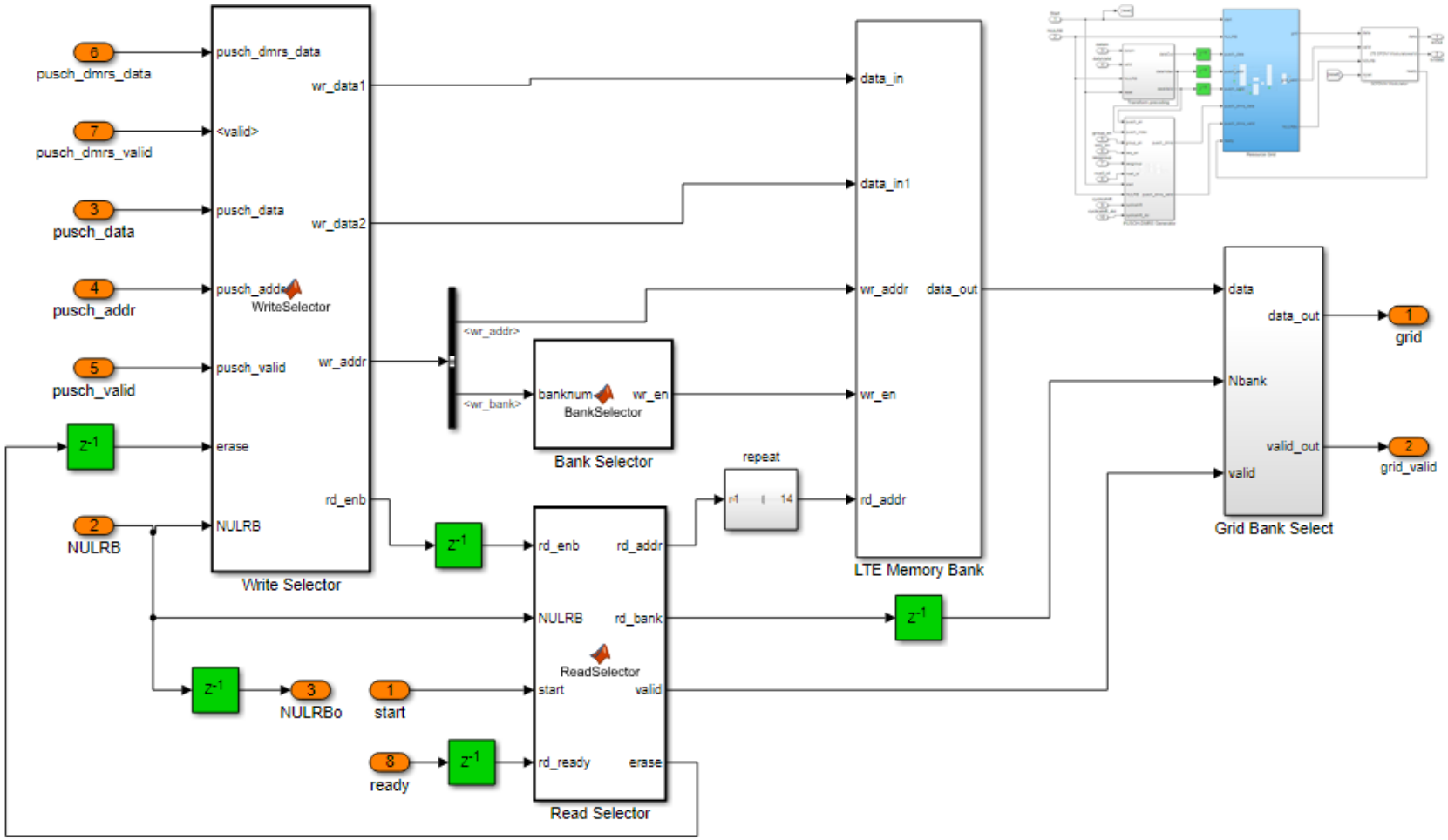

Figure 3.9: Resource grid block.

3.4.5 SC-FDMA Waveform Modulator

The *SC-FDMA Waveform Modulator*, shown in figure 3.10, has processing blocks coming from the *OFDM Modulator* block, present in the LTE HDL Toolbox library. The *OFDM symbol formation* in the original block was modified to accomodate for the SC-FDMA symbols. As the *OFDM symbol formation* discards the subcarrier located at the DC component in the symbol, the *SC-FDMA symbol formation* does not discard this subcarrier. Then, with the *Half Subcarrier IFFT* block added, between the *IFFT* and *FFT shift (in time domain)* blocks, the subcarriers are shifted by half the subcarrier spacing. These are the modifications made, while having the blocks used by the original *OFDM Modulator* block, to generate a SC-FDMA baseband signal. With these block modifications, the *SCFDMA Waveform Modulator* has the following processing blocks:

- *SC-FDMA symbol formation*: determines the number of active and inactive subcarriers, based on the number of resource blocks given by `NULRB`. It generates a `ready` signal whenever the block is ready to accept data from the *Resource Grid* block. This signal remains *high* until it has received a valid SC-FDMA symbol to fill in the active subcarriers.

- *IFFT*: converts a frequency-domain signal to a time-domain signal. LTE supports six standard bandwidth options: 1.4 MHz, 3 MHz, 5 MHz, 10 MHz, 15 MHz, and 20 MHz. These bandwidth options require IFFT sizes of 128, 256, 512, 1024, and 2048. The block uses a 2048 IFFT size, which corresponds to the maximum bandwidth of LTE, which is 20 MHz. The IFFT size is configured to the highest IFFT size to generate single hardware that supports all LTE bandwidth options [20].

- *Half Subcarrier IFFT*: shifts the IFFT-processed data by half the subcarrier spacing. It multiplies each complex-valued data sample by one of the stored complex values inside the *Half Subcarrier IFFT* block, which performs the desired half subcarrier spacing shift. This puts the DC component between two equally spaced subcarriers.

- *FFT shift (in time domain)*: performs a FFT shift in the time domain by multiplying each sample by a factor of $(-1)^n$. Each of the samples are either multiplied by +1 or -1.

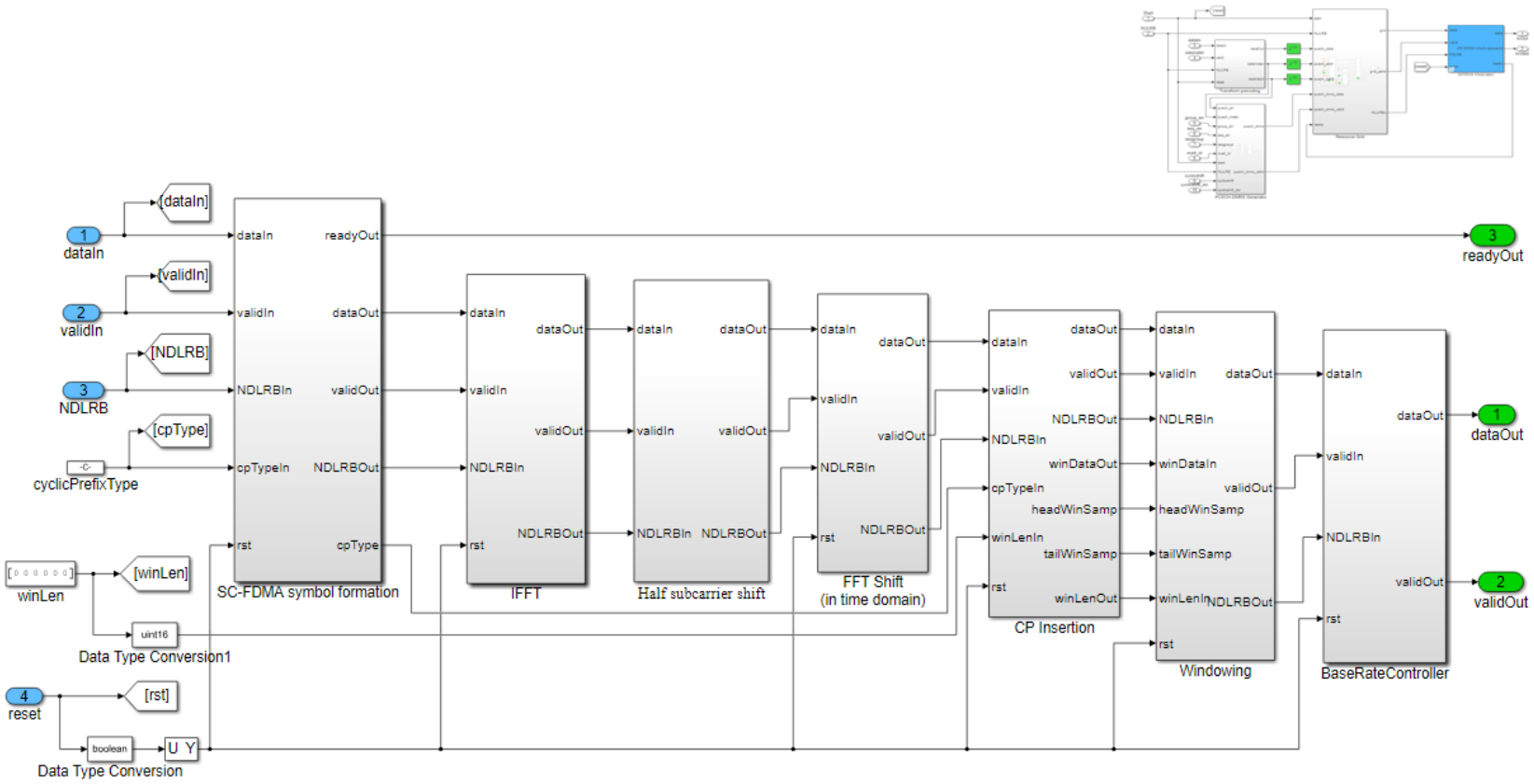

Figure 3.10: SC-FDMA Waveform Modulator block.

- *CP Insertion*: it adds the last samples of a SC-FDMA symbol as a cyclic prefix, for each SC-FDMA symbol. The number of samples to be inserted as a cyclic prefix is 160 for the first symbol and 144 for the remaining symbols in a slot, since the SC-FDMA block size from the IFFT process is 2048 samples.

- *Windowing*: this feature is optional. If enabled, it performs the windowing function, which smooths the edges of the symbol with cyclic prefix inserted, reducing out of band emissions on each SC-FDMA symbol. The windowed symbols are then overlapped, ensuring the time between symbols is maintained.

- *BaseRateController*: generates output data at a sample rate matching the `NULRB` parameter, in respect to the number of resource blocks. The output sample rate, at a given number of resource blocks, can be seen in the parameter *Sampling Frequency*, at table 2.7.

Taking into consideration the LTE Uplink transmitter designed in Simulink, the next phase is to show results from simulation in MATLAB and in VHDL testbench, validation of the system and resource utilization estimates.

Chapter 4

Results

4.1 Introduction

The structure of this chapter is to show and analyze the results achieved from two simulations of Uplink-Tx system: MATLAB simulation and VHDL testbench. The objective is to compare and validate the data produced by both simulations with the Wireless Waveform Generator data, for multiple resource block allocations. In order to validate the uplink data produced by the generator, a series of procedures were made, as shown in figure 4.1.

The Wireless Waveform Generator, from the MATLAB toolbox, can generate uplink resource grid data and SC-FDMA waveform, with various parameters that can be specified by the user, as illustrated in figure 4.2. A SC-FDMA waveform data in 1 frame was tested in a Vector Signal Analyzer (VSA) and it proves that the PUSCH and PUSCH-DMRS data are detected correctly, as seen in figure 4.3. Because of this, the generator from the MATLAB toolbox proves to be a viable tool for comparison and validation of data output at the Uplink-Tx system.

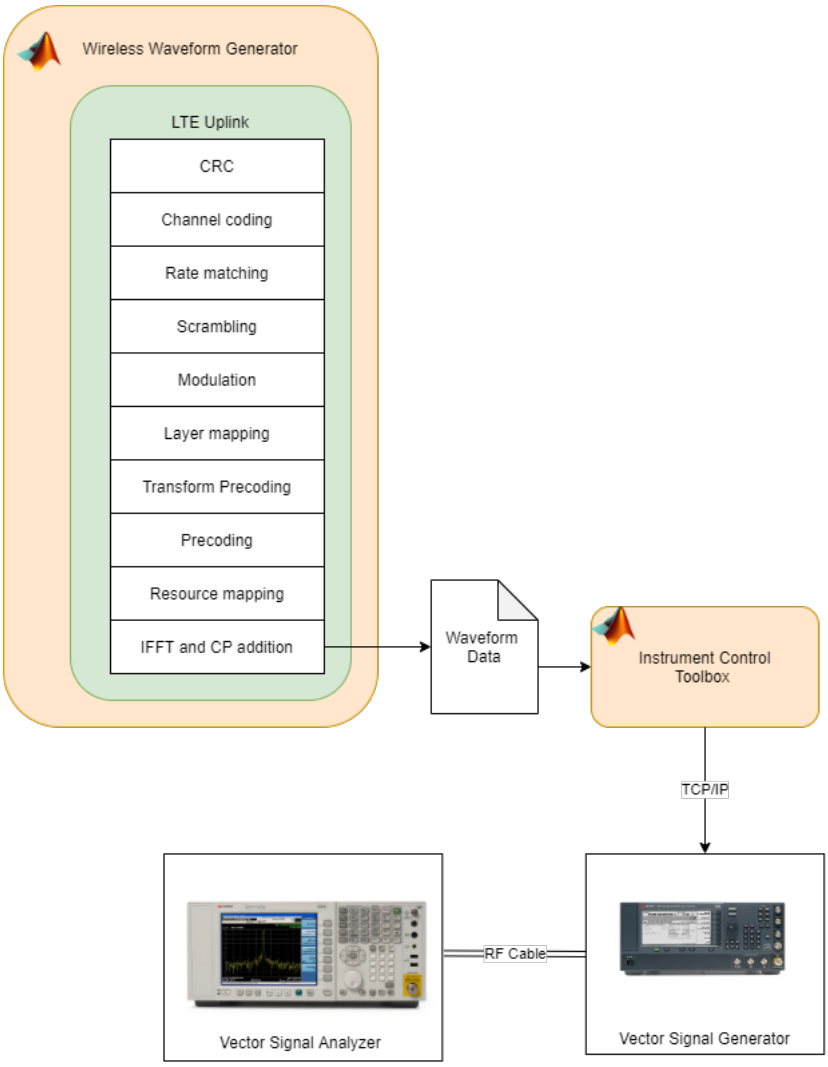

Figure 4.1: Procedural diagram of Wireless Waveform Generator test.

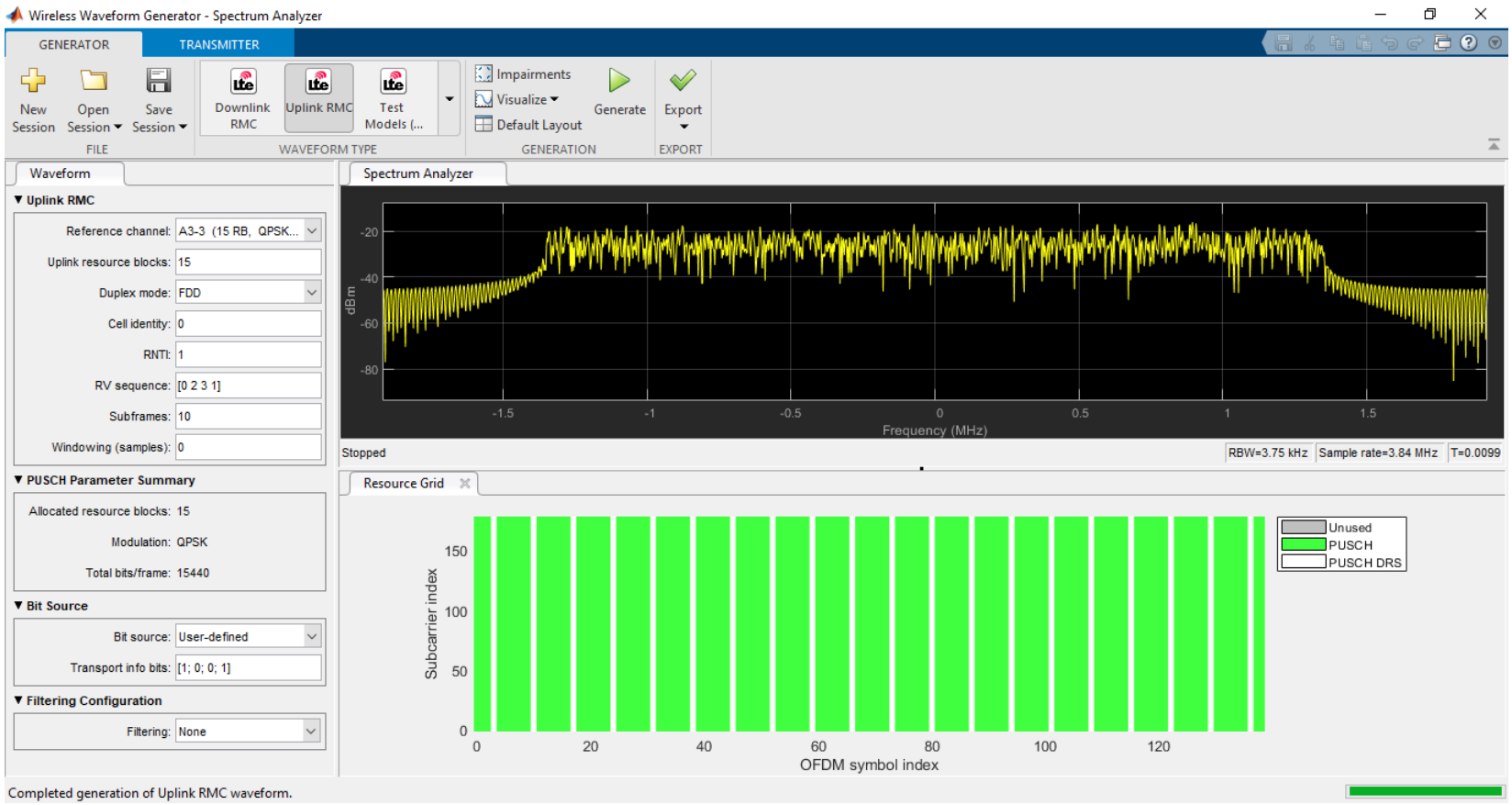

Figure 4.2: Waveform Generator from MATLAB Toolbox.

Figure 4.3: VSA with received uplink signal, produced by the Waveform Generator from MATLAB Toolbox.

4.2 MATLAB/Simulink Simulation

In order to test the Uplink-Tx system in Simulink, the Wireless Waveform Generator in MATLAB toolbox generates resource grid data according to uplink parameters from upper layers. The parameters are the number of resource blocks for PUSCH and PUSCH-DMRS, the PUSCH modulation (QPSK, 16QAM, and 64QAM), the bandwidth for the resource grid, the duplex mode (FDD or TDD), cell identity, number of subframes for simulation, the number of samples for windowing and number o transport bits for PUSCH, as shown in figure 4.2. From the resource grid data generated, it is extracted the PUSCH data and PUSCH-DMRS parameters. The extracted PUSCH data goes through a Transform De-Precoding process, that is performing IFFT followed by a multiplication with a factor of $\sqrt{M_{sc}^{PUSCH}}$, with M_{sc}^{PUSCH} representing the size of PUSCH data in a symbol, in number of subcarriers. This process reverts the PUSCH data, processed after Transform Precoding, back into symbol-modulated PUSCH data. After this, the data is upsampled up to either 1200 or 1800 samples, depending on the bandwidth for the resource grid, in number of resource blocks. The symbol-modulated PUSCH data and PUSCH-DMRS parameters serve as inputs for the Uplink-Tx system. The Uplink-Tx system generates SC-FDMA waveform data at the output, which is then demodulated back to the form of resource grid data. From this, the PUSCH-DMRS data is extracted, while PUSCH data goes through another Transform De-precoding process. Finally, it is shown the constellation diagram from PUSCH and PUSCH-DMRS data, which is then compared to the Transform De-Precoding data from the resource grid generated by the Wireless Waveform Generator. The figure 4.4 shows the procedural diagram to obtain results from Simulink.

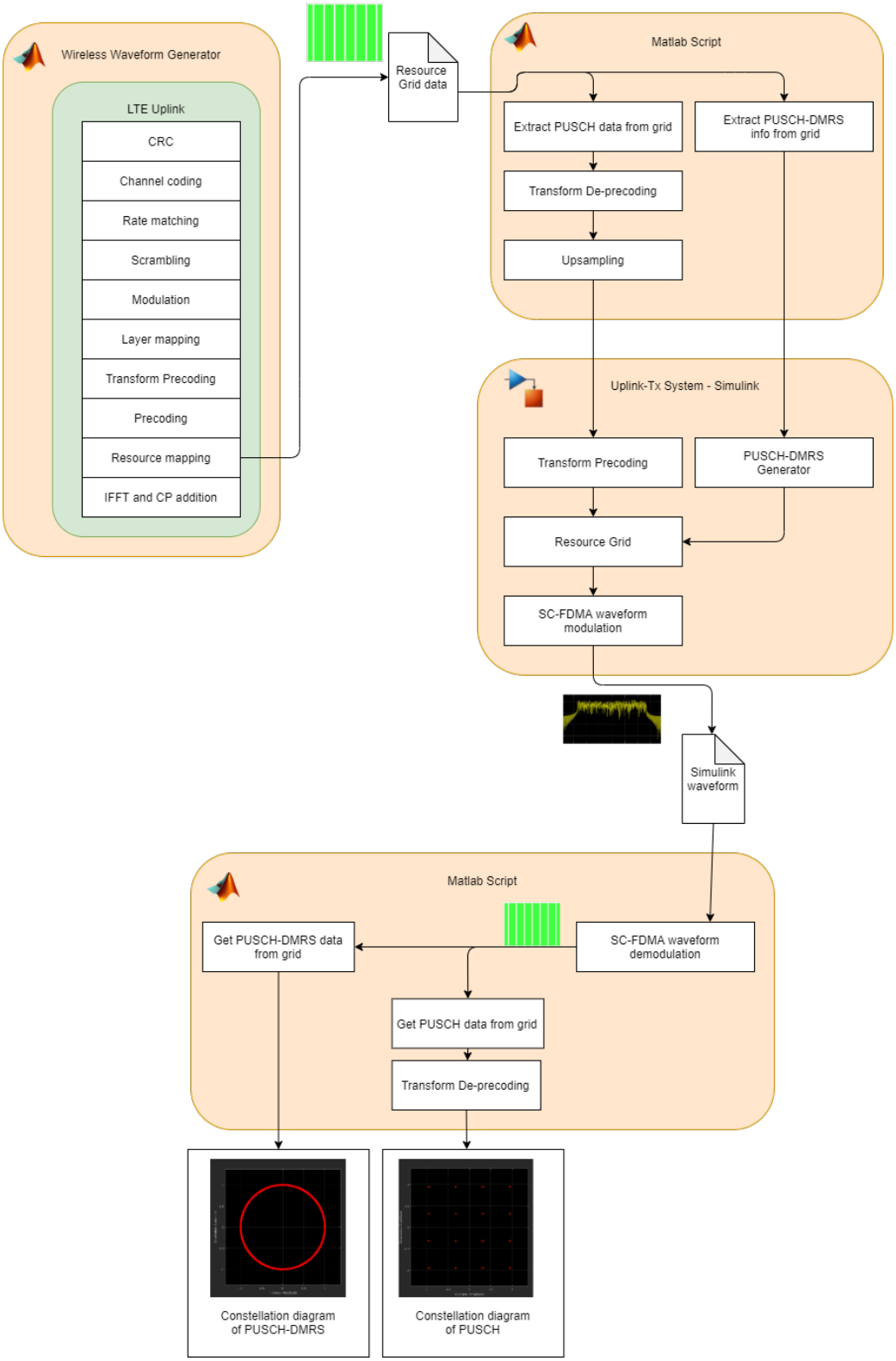

Figure 4.4: Procedural diagram of the results from Simulink.

The Uplink-Tx system is simulated with PUSCH data input, modulated in 16QAM, with 30720 samples for each subframe. As each subframe of data has 1 ms of duration, the sampling frequency for the system in Simulink is 30.72 MHz. Besides PUSCH data, the Uplink-Tx also receives as input the number of resource blocks for PUSCH and resource grid bandwidth, in number of resource blocks, which can be 6, 15, 25, 50, 75 and 100. The simulation time is equal to 30720*(number of subframes)+14000 samples. The delay between the first valid input and first valid output is 12841, 12949, 11269, 11569, 13669 and 12169 samples, for 6, 15, 25, 50, 75 and 100 resource blocks, respectively. Since the largest delay is 13669 samples, 14000 samples are added to the simulation time to compensate for all delays.

The figures in the Appendices section, specifically Annex B, show the constellation diagrams with 1 subframe of data of PUSCH and PUSCH-DMRS data at an ideal receiver. For the results, it was considered two different scenarios of received signal, one without noise and another with white gaussian noise added, with a Signal-to-Noise Ratio (SNR) of 20 dB. The figures are from simulations with 1.4, 3, 5, 10, 15 and 20 MHz bandwidth, corresponding to resource blocks of 6, 15, 25, 50, 75 and 100, respectively. Each constellation diagram displays the ideal constellation points in red, while the signal obtained at the receiver is displayed in yellow. On the side, it shows the Error Vector Magnitude (EVM) values, which is the difference between the symbols obtained at the receiver and the ideal symbols. The "RMS EVM" represents the average Root Mean Square (RMS) of EVM and "Avg MER" the Modulation Error Ratio (MER) between the average power of the signal and the average power of the error vector. The EVM values are normalized by the average constellation power. From the constellation diagrams, it can be seen that, even with added noise, the information present in the data picked up by the receiver is not compromised.

In the scenario where the Uplink-Tx processes the maximum possible amount of samples, that maximum being 1 frame of data with 20 MHz bandwidth, the average RMS EVM for PUSCH and PUSCH-DMRS data does not exceed 12.5%, as observed in figures 4.5, 4.6, 4.7 and 4.8. This value is the minimum requirement for average RMS EVM level defined by 3GPP, on section 6.5.2.1.1 in [8], for a 16QAM modulation.

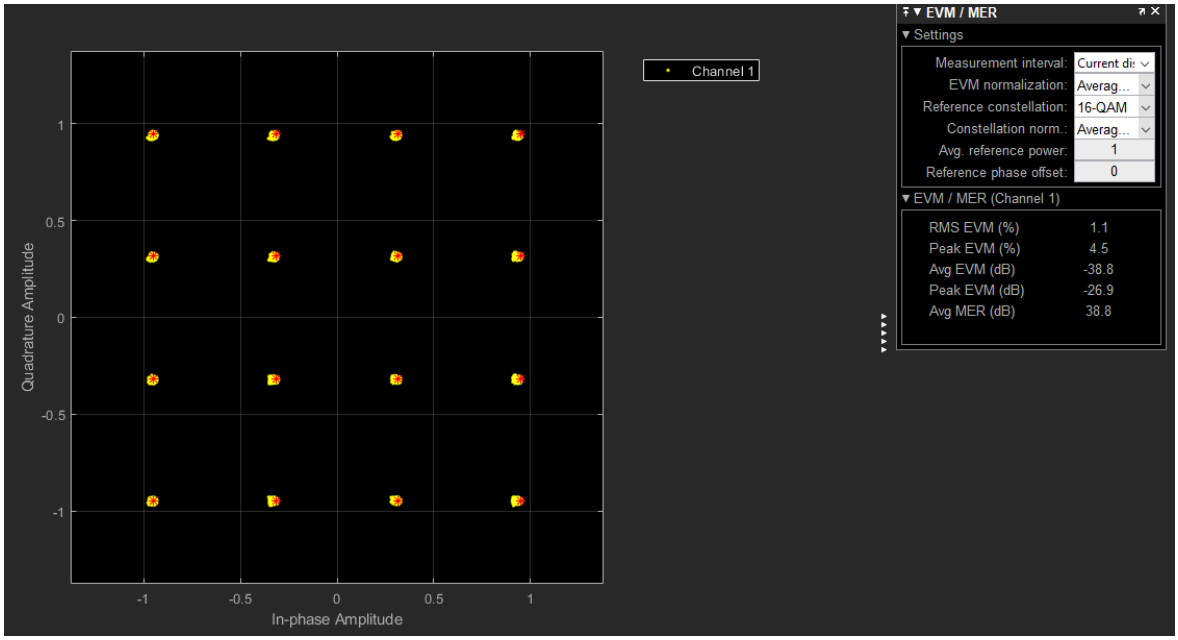

Figure 4.5: Constellation diagram of PUSCH, from 1 frame of data, for 20 MHz bandwidth, on Simulink.

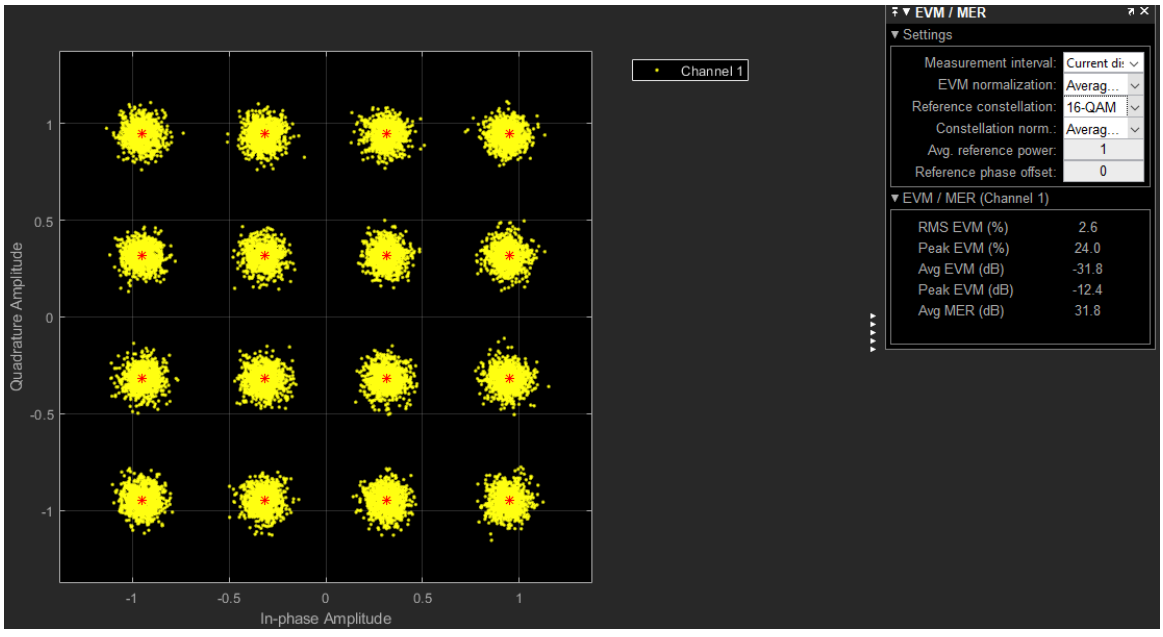

Figure 4.6: Constellation diagram of PUSCH with added noise, from 1 frame of data, for 20 MHz bandwidth, on Simulink.

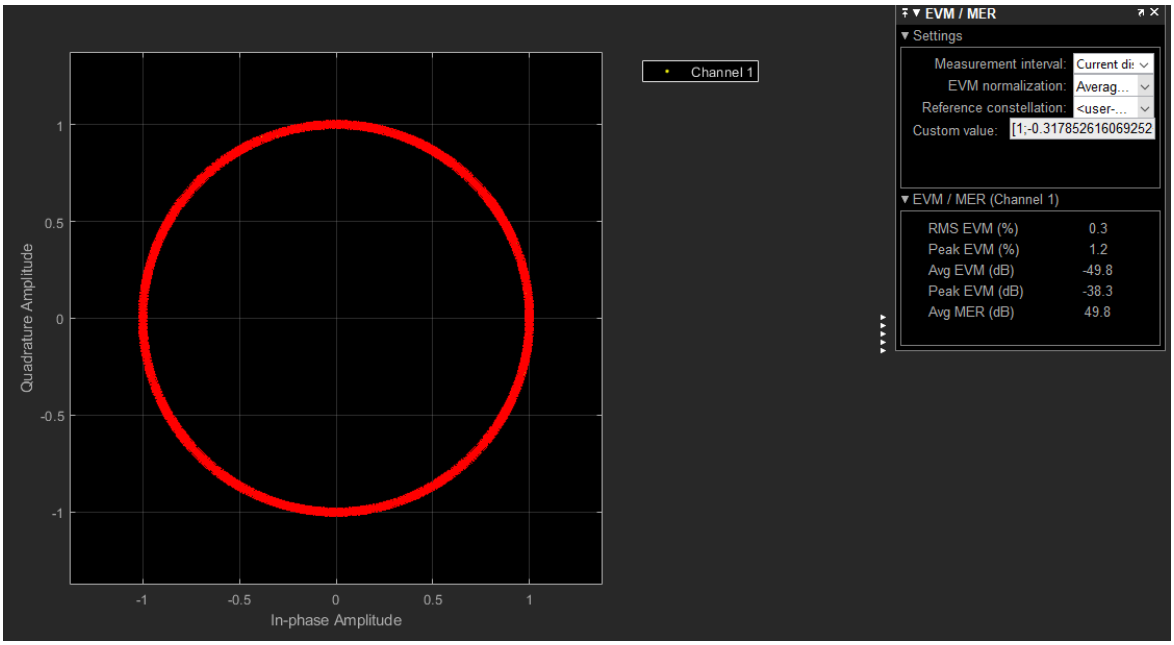

Figure 4.7: Constellation diagram of PUSCH-DMRS, from 1 frame of data, for 20 MHz bandwidth, on Simulink.

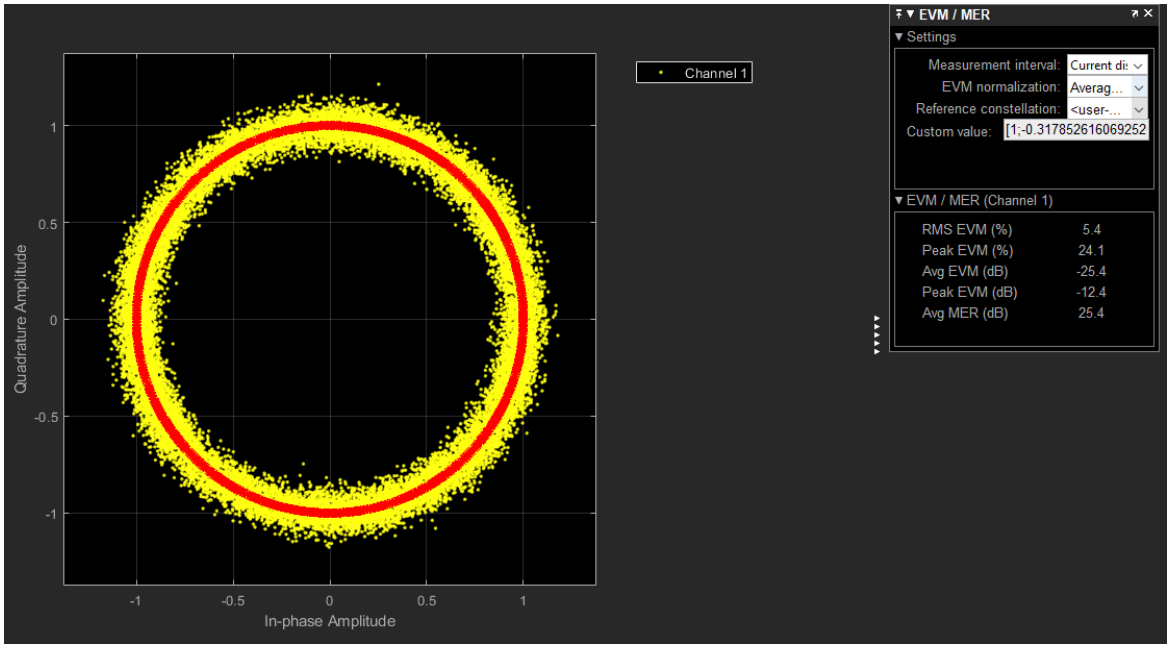

Figure 4.8: Constellation diagram of PUSCH-DMRS with added noise, from 1 frame of data, for 20 MHz bandwidth, on Simulink.

4.3 VHDL Testbench

The procedure for VHDL testbench simulation begins with the Wireless Waveform Generator, from MATLAB toolbox, generating resource grid data according to uplink parameters from upper layers. From the resource grid data generated, it is extracted the PUSCH data and PUSCH-DMRS parameters. The extracted PUSCH data goes through a Transform De-Precoding process and then through upsampling, ending up on a symbol-modulated PUSCH data with zeroes. This PUSCH data and the PUSCH-DMRS parameters are stored in different .dat files, on hexadecimal format. The VHDL testbench reads the .dat files of PUSCH data and PUSCH-DMRS parameters, runs simulation for the Uplink-Tx system and stores the resulting SC-FDMA data in a different .dat file. The SC-FDMA data is then read and converted to fixed-point complex data, proceeding to the SC-FDMA waveform demodulation into resource grid data, getting PUSCH and PUSCH-DMRS data. The PUSCH data goes through a Transform De-precoding, getting the desired PUSCH data in symbol-modulated form. This and the PUSCH-DMRS data are then shown in constellation diagrams, to compare with the Transform De-Precoding data processed from the resource grid data of the Wireless Waveform Generator. The figure 4.9 shows the procedural diagram to obtain results from VHDL testbench.

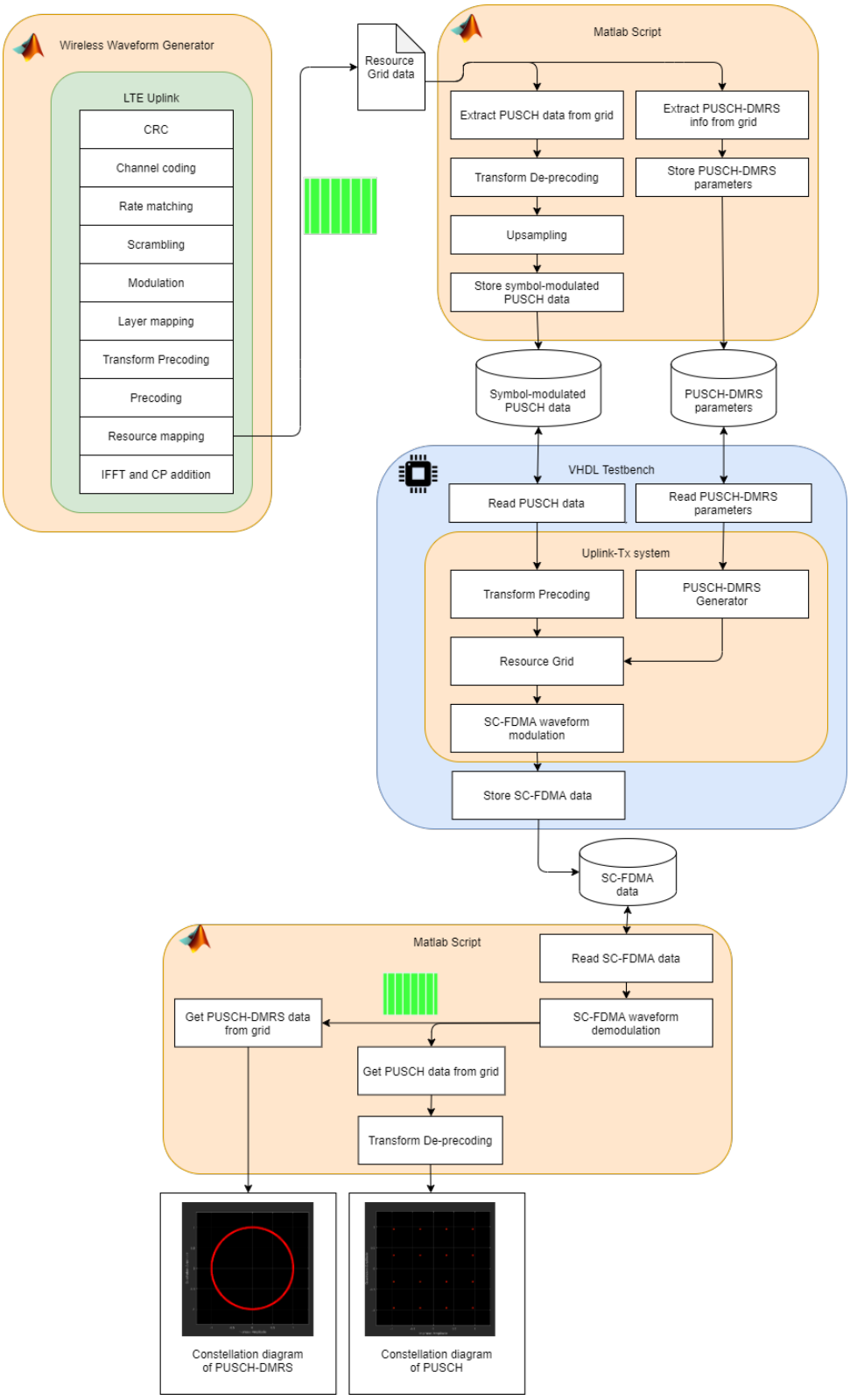

Figure 4.9: Procedural diagram of the results from VHDL testbench.

As presented in figure 4.10, the modulated PUSCH and PUSCH-DMRS data are separated into their real and imaginary components, corresponding to `dataIn_re` and `dataIn_Im` signals, respectively. These two variables are unitialized until the `clk_enable` is set to *high*. The first input sample comes up at 32 ns. The `txOut_re_addr` is the number of samples stored in the output .dat file, in hexadecimal format, which increases as the `clk_enable` is set to *high* and the `clk` signal rises. This testbench is tested for 6 resource blocks, indicated by the `rawData_NULRB` parameter.

The first valid output samples, indicated by the `txValid` signal, show up at 128560 ns. The delay, in this case, between the first input samples and the first output samples, is equal to 128560 - 32 = 128528 ns, or approximately 12853 samples. This matches with the value of `txOut_re_addr` in decimal format, at the first valid output sample: 3224_{16}+1=12852_{10}+1=12853.

The testbench is programmed to process 44720 samples, that is the 30720 samples in a subframe of data plus 14000 samples, to compensate for the delay between input and output, for all cases of different resource blocks. However, for the case of 6 resource blocks, the simulation stops when the `txOut_re_addr` reaches the value 43720, or AAC8 in hexadecimal format. The `txValid_done` marks the end of the simulation, at 437230 ns, or 437.230 us.

In the case where the Uplink-Tx system in testbench processes 1 frame of data with 20 MHz bandwidth, the constellation diagrams obtained in an ideal receiver, shown in figures 4.11, 4.12, 4.13 and 4.14, presents an average RMS EVM for PUSCH and PUSCH-DMRS data that does not exceed 12.5%. Also, even with added noise, making an SNR of 20 dB, the information present in the data received is not compromised.

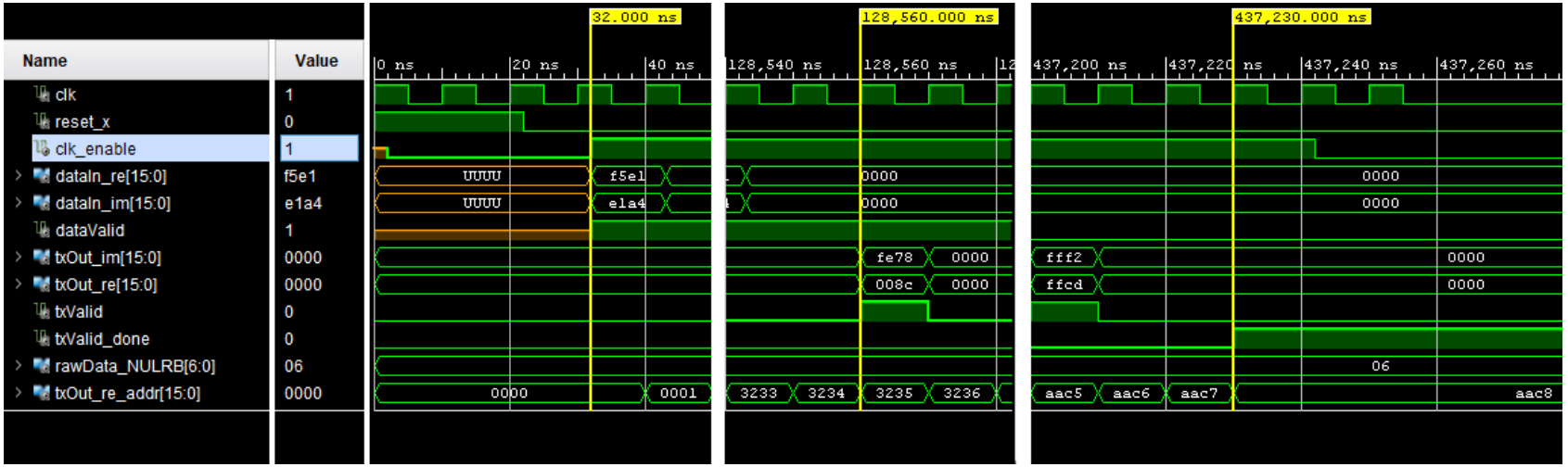

Figure 4.10: Testbench timeline, at the start of the simulation, at the moment of the first output sample and at the end of the simulation.

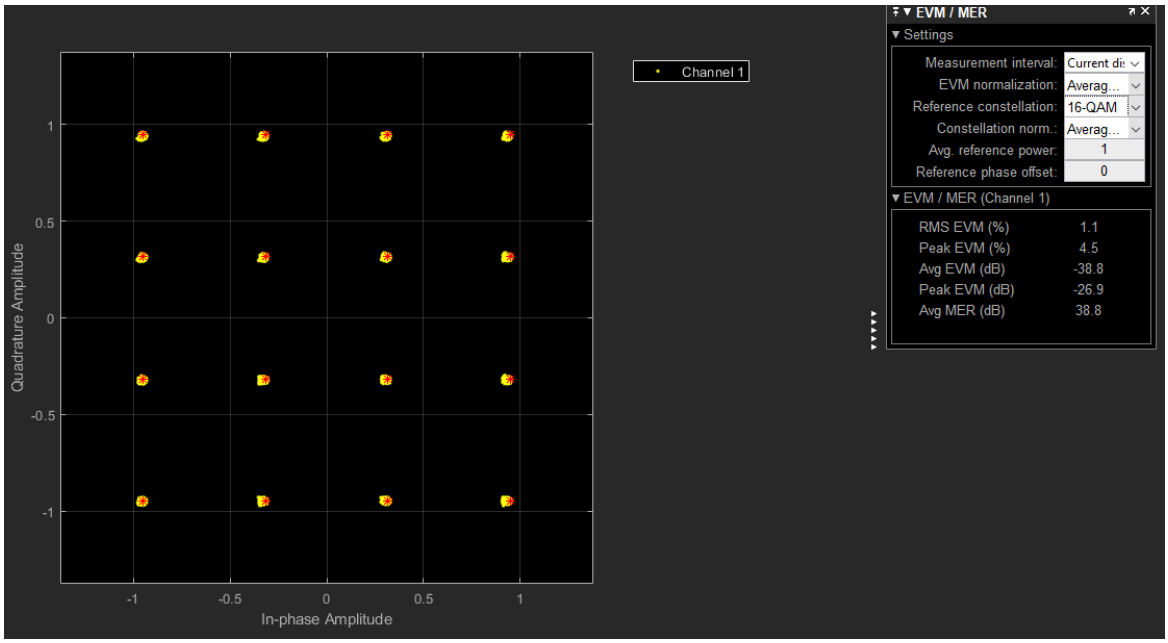

Figure 4.11: Constellation diagram of PUSCH, from 1 frame of data, for 20 MHz bandwidth, on testbench.

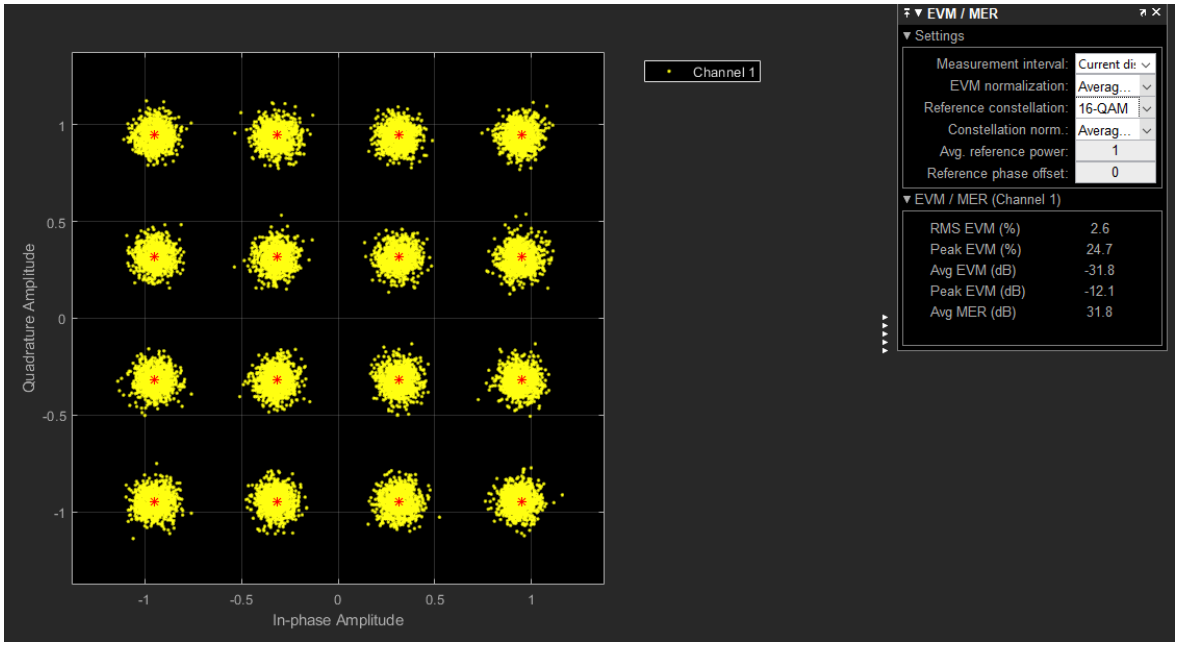

Figure 4.12: Constellation diagram of PUSCH with added noise, from 1 frame of data, for 20 MHz bandwidth, on testbench.

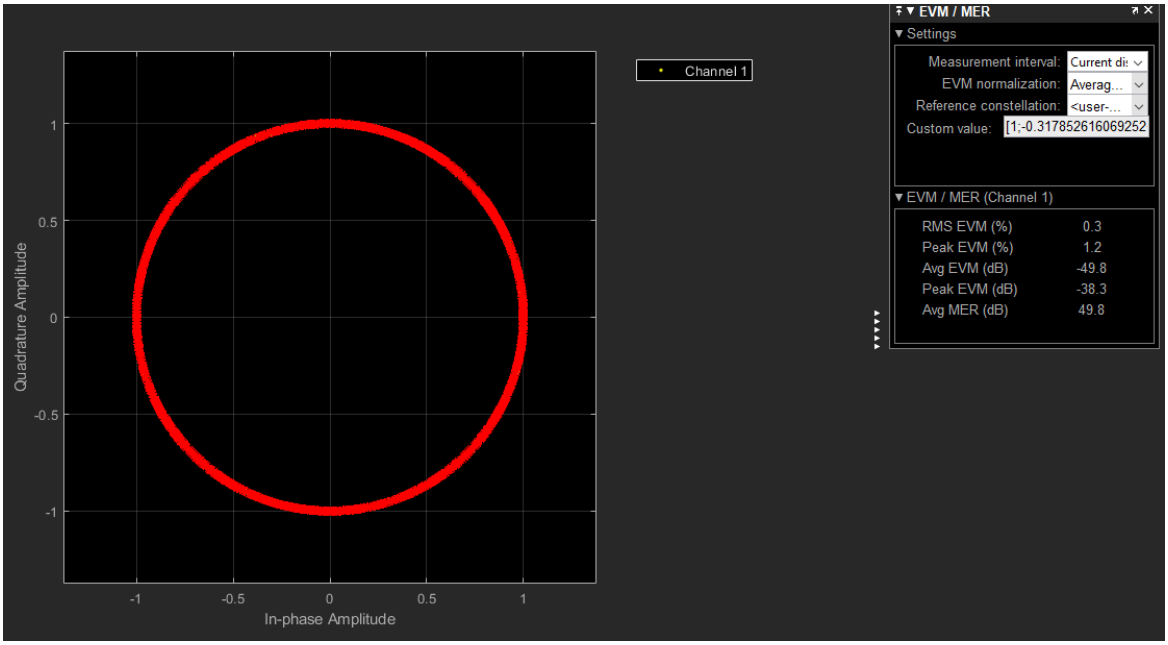

Figure 4.13: Constellation diagram of PUSCH-DMRS, from 1 frame of data, for 20 MHz bandwidth, on testbench.

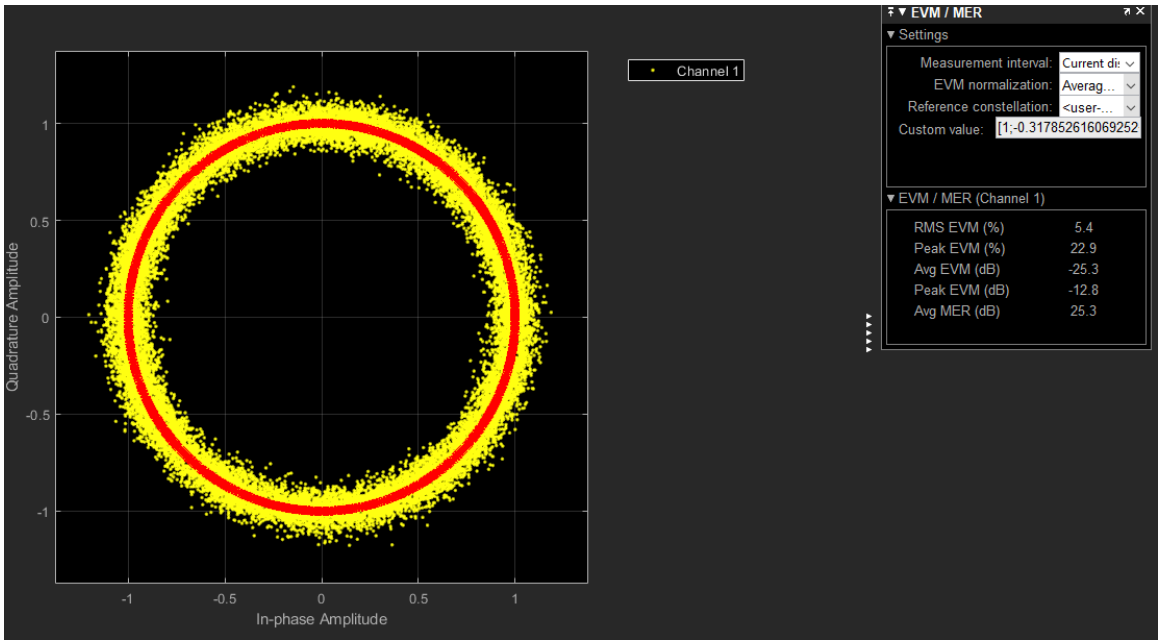

Figure 4.14: Constellation diagram of PUSCH-DMRS with added noise, from 1 frame of data, for 20 MHz bandwidth, on testbench.

4.4 Resource utilization estimates

The HDL Coder reported resource utilization estimates observed in table 4.1, for the Uplink-Tx system. It also shows resource utilization estimates after VHDL synthesis in table 4.2. This table shows the available resources on different FPGAs, chosen as examples, to show how much of the available resources the Uplink-Tx system can be in use.

Table 4.1: Estimation of resource utilization obtained from HDL coder.

Type of resource	**Num. of resources**
Multipliers	77
Adders/Subtractors	675
Registers	10772
Total 1-Bit Registers	118640
RAMs	93
Multiplexers	1489
I/O Bits	100
Static Shift operators	107

Table 4.2: Resource utilization report after VHDL synthesis, with resources available for different FPGAs [21] [22].

	Estimation post-synthesis	**XC7A100TCSG324 resources used (%)**	**ZU9EG-2FFVB1156 resources used (%)**
Slice LUT	27153	43	10
Slice Registers	15926	13	3
RAM Blocks (36Kbits)	60.5	45	7
DSP Slices	82	34	3

With the LTE Uplink transmitter system validated from the results obtained and resource utilization estimates, the next point of this dissertation are the conclusions about the developed work and suggestions for future works that can be developed, based on the work of this dissertation.

Chapter 5

Conclusion and Future work

5.1 Conclusion

The purpose of this dissertation was to develop in VHDL a 4G Uplink-Tx physical layer system. The system modelling in Simulink of such system, in order to synthesize it in VHDL, was achieved for PUSCH data. This achievement was made possible with the Wireless Waveform Generator in the MATLAB toolbox, as it ensured as a good comparison model, while developing the system in Simulink and comparing the results from the system in Simulink and in VHDL testbench.

Initially, this document shows an introduction to 4G LTE, with the motivation explaining the deployment of 5G network and explaining the already existing 4G LTE networks, the coexistence between these two networks and finishing off with the objectives and structure of the work for this dissertation. After that, it is shown a more detailed information about LTE Uplink subsystem, in order to understand the concepts in the modelling of the physical layer system and the obtained results. Furthermore, this document explains each of the processing blocks that compose the uplink transmitter system developed, designed in a way that HDL Coder could synthesize those blocks in VHDL. Lastly, the results are shown from simulations in Simulink and VHDL testbench, with resource allocation estimates. Through the results obtained and analysis made, it is possible to verify and validate the uplink transmitter system developed, fulfilling the objectives defined for this dissertation.

This developed system is still far from being a fully functional uplink transmitter system, with support for all the uplink channels and reference signals, mentioned in this dissertation.

5.2 Future work

This section shows some proposed future works which can be based on the developed LTE Uplink-Tx system.

Implementation in FPGA

While the results show that it can be possible to implement the VHDL into the FPGA hardware, it does not show the results of implementation in a FPGA and how it processes the inputs fed into it, even though the focus of this dissertation being the development in VHDL of a transmitting system capable of processing uplink data. This proposed work should be considered before the realization of the other two proposed works, as it might prove itself as a study on how the FPGA works with the transmitting system developed in this thesis, with the used processing blocks.

Integration with SRS and PUCCH and PRACH channels

The Uplink-Tx physical layer system needs to be a system capable of processing information from all existing physical uplink channels - PUSCH, PUCCH and PRACH - and reference channels - DMRS and SRS.

While MATLAB toolbox provides a waveform generator to generate LTE Uplink signal to be transmitted, its waveform is generated from a resource grid with only PUSCH information mapped into it. MATLAB toolbox also provides specific built-in functions, but it would involve putting them together by hand, to function as an uplink system. The signal processed by a system with built-in functions in MATLAB would require being tested in VSA, in order for it to be a valid and comparable system. The built-in functions can provide a reference for generating the intended signal, but the validation must be made on a VSA.

MIMO System

The designed Uplink-Tx transmitter for this dissertation is a single-input, single-output system. For this work, the transmitter developed needs the Precoding processing function, as seen on figure 1.6, for PUSCH and PUSCH-DMRS. The resource grid processing function needs more memory space, varying with the number of transmitting antennas. Also, the

uplink transmitter would need more SC-FDMA baseband signal generators for each antenna. Developing a MIMO System does not need to be a separated work from the other previous suggested works.

Bibliography

[1] H. Zarrinkoub. *Understanding LTE with MATLAB: From Mathematical Modeling to Simulation and Prototyping*. 1st. Wiley Publishing, 2014.

[2] 3GPP. *5G; NR; Overall description; Stage-2 (3GPP TS 36.300 version 10.3.0 Release 10)*. 2011

[3] 3GPP. *5G; NR; Overall description; Stage-2 (3GPP TS 38.300 version 15.3.1 Release 15)*. 2018

[4] S. Shew. *Transport network support of IMT-2020/5G*. 2018

[5] Y. Chen, X. Lagrange. *Architecture and Protocols of EPC-LTE with relay*. 2013, pp.25. ffhal-00830621

[6] ITU-R Telecommunication standardization sector of ITU. *5G wireless fronthaul requirements in a passive optical network context*. 2019

[7] 3GPP. *Technical Specification Group Radio Access Network; Study on CU-DU lower layer split for NR; (Release 15)*. 2017

[8] 3GPP Terrestrial Radio Access (E-UTRA). *User Equipment (UE) radio transmission and reception (3GPP TS 36.101 version 15.3.0 Release 15)*. 2018

[9] 3GPP Terrestrial Radio Access (E-UTRA). *Physical channels and modulation (3GPP TS 36.211 version 10.1.0 Release 10)*. 2011

[10] 3GPP Terrestrial Radio Access (E-UTRA). *Physical layer procedures (3GPP TS 36.213 version 10.1.0 Release 10)*. 2011

[11] S. Sesia, I. Toufik, M. Baker. *LTE – The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing. 2009

[12] Y. He, Y. Su, E. Dutkiewicz, X. Huang and J. Shi. *An efficient implementation of uplink baseband signal generator in LTE UE transmitters*. 2012 International Symposium on Communications and Information Technologies (ISCIT), Gold Coast, QLD, 2012, pp. 444-448. Link: `https://ieeexplore.ieee.org/abstract/document/6380939`. Note: Requires IEEE account or Institutional Sign In for access.

[13] N. Guo. *Implementation Aspects of 3GPP TD-LTE*. 2009. Link: `https://www.diva-portal.org/smash/get/diva2:234603/FULLTEXT01.pdf`

[14] P. Cunha. *Implementing the SC-FDMA Transmission Technique Using the GNURadio Platform*. 2014. Link: `https://run.unl.pt/bitstream/10362/12251/1/Cunha_2014.pdf`

[15] Mathworks. *Simulink*. Link: `https://www.mathworks.com/help/simulink`

[16] Mathworks. *HDL Coder*. Link: `https://www.mathworks.com/help/hdlcoder`

[17] Mathworks. *LTE HDL PBCH Transmitter*.
Link: `https://www.mathworks.com/help/releases/R2019a/ltehdl/examples/lte-hdl-transmitter.html`

[18] J. Löfgren, P. Nilsson. *On Hardware Implementation of Radix 3 and Radix 5 FFT Kernels for LTE systems*. 2011

[19] J. Rio. Link: `https://numericalrecipes.wordpress.com/2009/05/29/the-cooley-tukey-fft-algorithm-for-general-factorizations/`

[20] Mathworks. *OFDM Modulator*. Link: `https://www.mathworks.com/help/releases/R2019a/ltehdl/ref/ofdmmodulator.html`

[21] Xilinx. *7 Series FPGAs Data Sheet: Overview*. September 8, 2020. Link: `https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf`

[22] Xilinx. *Zynq UltraScale+ MPSoC Data Sheet: Overview*. October 2, 2019. Link: https://www.xilinx.com/support/documentation/data_sheets/ds891-zynq-ultrascale-plus-overview.pdf

Appendices

Annex A - LTE and LTE-Advanced operating bands.

Table A.1: LTE and LTE-Advanced operating bands on FDD Mode (extracted from [8]).

E-UTRA operating band	**Uplink (UL) operating band (MHz)**	**Downlink (DL) operating band (MHz)**	**Duplex Mode**
1	1920 - 1980	2110 - 2170	FDD
2	1850 - 1910	1930 - 1990	FDD
3	1710 - 1785	1805 - 1880	FDD
4	1710 - 1755	2110 - 2155	FDD
5	824 - 849	869 - 894	FDD
6	830 - 840	875 - 885	FDD
7	2500 - 2570	2620 - 2690	FDD
8	880 - 915	925 - 960	FDD
9	1749.9 - 1784.9	1844.9 - 1879.9	FDD
10	1710 - 1770	2110 - 2170	FDD
11	1427.9 - 1452.9	1475.9 - 1500.9	FDD
12	698 - 716	728 - 746	FDD
13	777 - 787	746 - 756	FDD
14	788 - 798	758 - 768	FDD
15	Reserved	Reserved	
16	Reserved	Reserved	
17	704 - 716	734 - 746	FDD
18	815 - 830	860 - 875	FDD
19	830 - 845	875 - 890	FDD
20	832 - 862	791 - 821	FDD
21	1447.9 - 1462.9	1495.9 - 1510.9	FDD
22	3410 - 3490	3510 - 3590	FDD
23	2000 - 2020	2180 - 2200	FDD

Continuation on next page

Table A.1: LTE and LTE-Advanced operating bands on FDD Mode (cont.).

E-UTRA operating band	**Uplink (UL) operating band (MHz)**	**Downlink (DL) operating band (MHz)**	**Duplex Mode**
24	1626.5 - 1660.5	1525 - 1559	FDD
25	1850 - 1915	1930 - 1995	FDD
26	814 - 849	859 - 894	FDD
27	807 - 824	852 - 869	FDD
28	703 - 748	758 - 803	FDD
29	N/A	717 - 728	FDD
30	2305 - 2315	2350 - 2360	FDD
31	452.5 - 457.5	462.5 - 467.5	FDD
32	N/A	1452 - 1496	FDD
72	451 - 456	461 - 466	FDD
Note: Band 6 is not applicable, as of release 10 and beyond. Band 23 is also not applicable, as of release 14 and beyond.			

Table A.2: LTE and LTE-Advanced operating bands on TDD Mode (extracted from [8]).

E-UTRA operating band	**Uplink (UL) operating band (MHz)**	**Downlink (DL) operating band (MHz)**	**Duplex Mode**
33	1900 - 1920	1900 - 1920	TDD
34	2010 - 2025	2010 - 2025	TDD
35	1850 - 1910	1850 - 1910	TDD
36	1930 - 1990	1930 - 1990	TDD
37	1910 - 1930	1910 - 1930	TDD
38	2570 - 2620	2570 - 2620	TDD
39	1880 - 1920	1880 - 1920	TDD
40	2300 - 2400	2300 - 2400	TDD
41	2496 - 2690	2496 - 2690	TDD
42	3400 - 3600	3400 - 3600	TDD
43	3600 - 3800	3600 - 3800	TDD
44	703 - 803	703 - 803	TDD
45	1447 - 1467	1447 - 1467	TDD
46	5150 - 5925	5150 - 5925	TDD
47	5855 - 5925	5855 - 5925	TDD
48	3550 - 3700	3550 - 3700	TDD
49	3550 - 3700	3550 - 3700	TDD
50	1432 - 1517	1432 - 1517	TDD
51	1427 - 1432	1427 - 1432	TDD
52	3300 - 3400	3300 - 3400	TDD

Annex B - Constellation diagram of PUSCH and PUSCH-DMRS at the receiver, from the transmitter on Simulink.

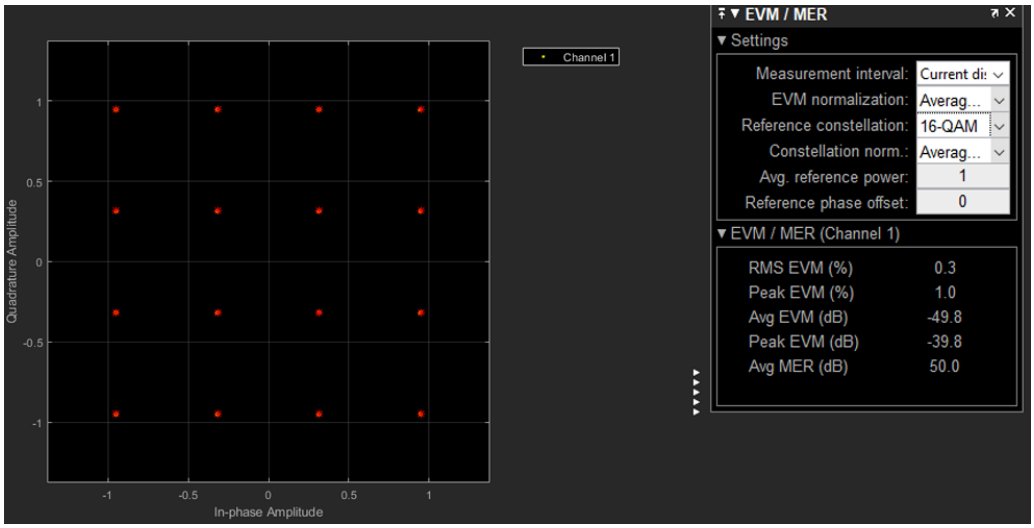

Figure B.1: Constellation diagram of PUSCH, from 1 subframe of data, for 1.4 MHz bandwidth.

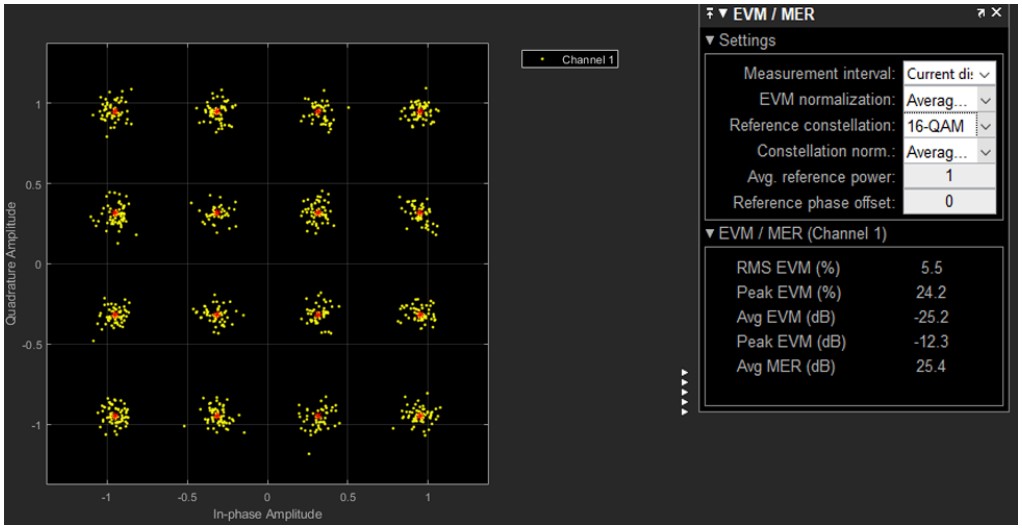

Figure B.2: Constellation diagram of PUSCH with added noise, from 1 subframe of data, for 1.4 MHz bandwidth.

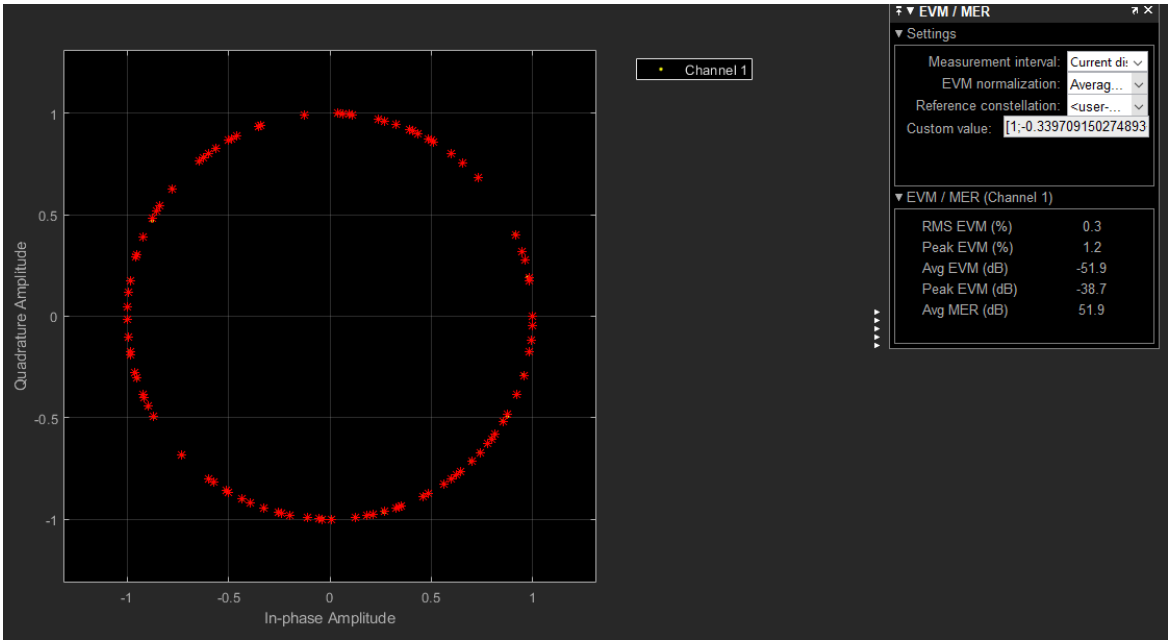

Figure B.3: Constellation diagram of PUSCH-DMRS, from 1 subframe of data, for 1.4 MHz bandwidth.

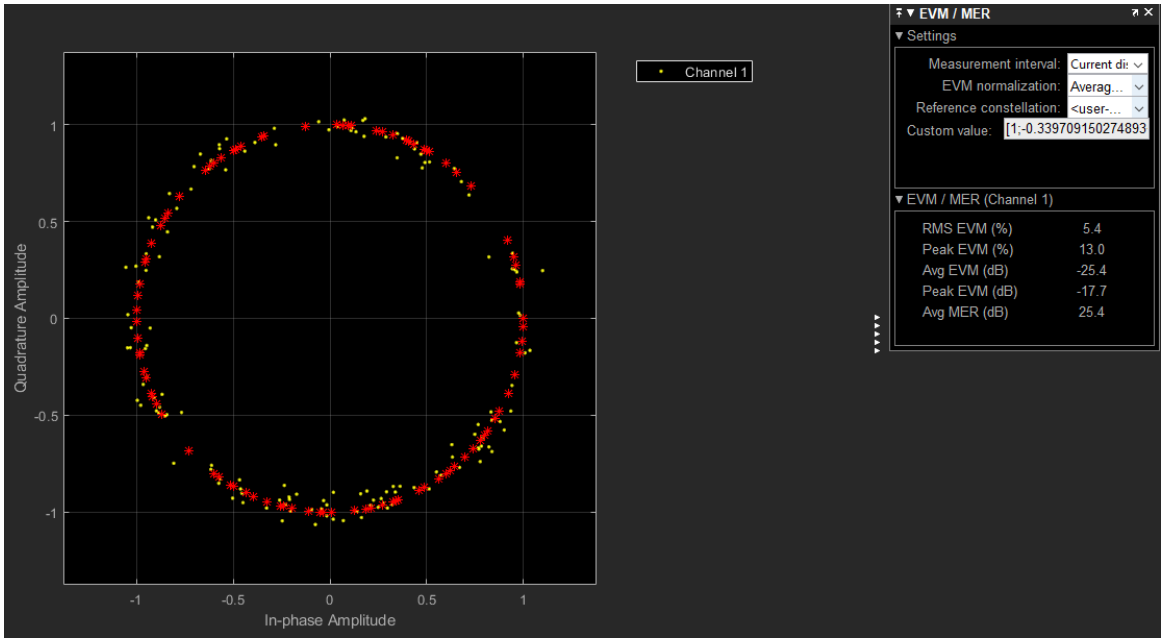

Figure B.4: Constellation diagram of PUSCH-DMRS with added noise, from 1 subframe of data, for 1.4 MHz bandwidth.

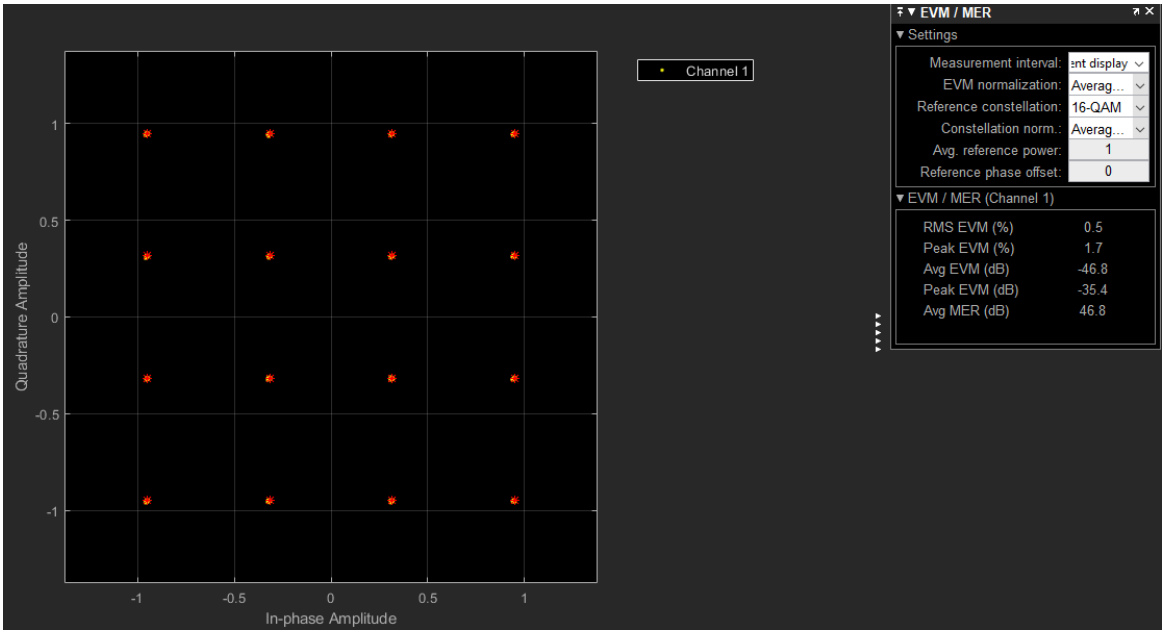

Figure B.5: Constellation diagram of PUSCH, from 1 subframe of data, for 3 MHz bandwidth.

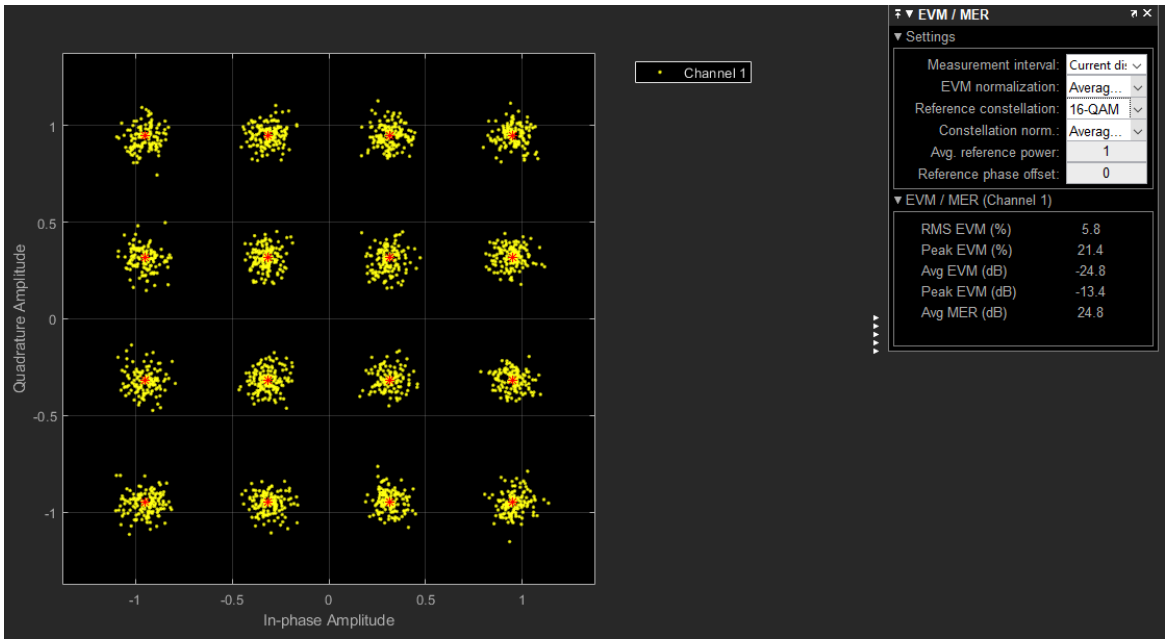

Figure B.6: Constellation diagram of PUSCH with added noise, from 1 subframe of data, for 3 MHz bandwidth.

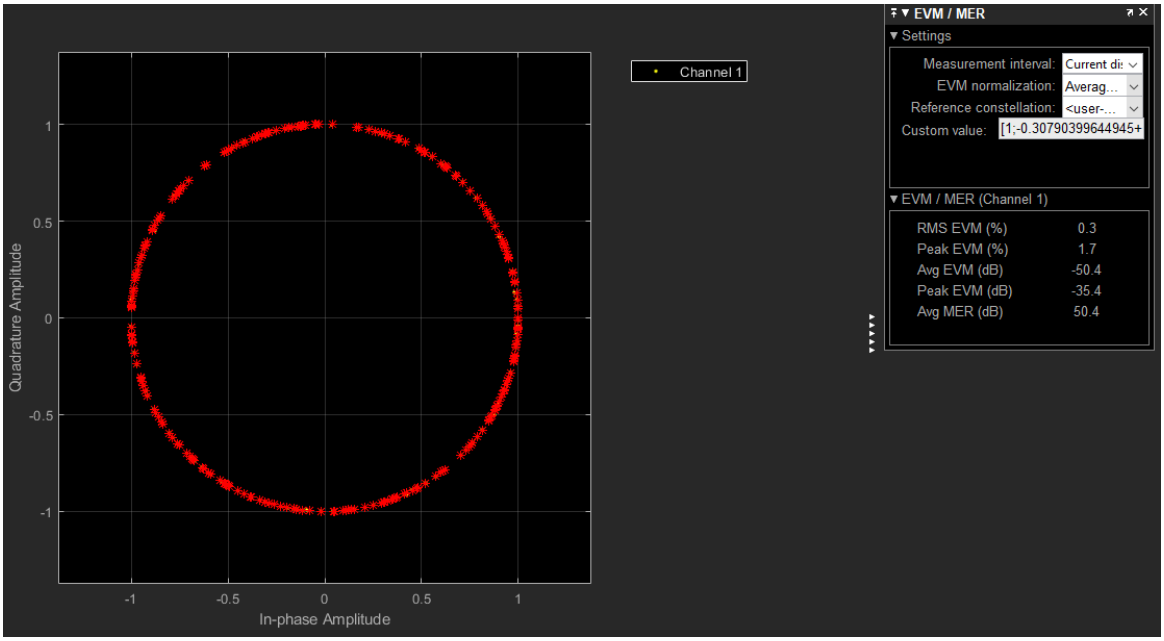

Figure B.7: Constellation diagram of PUSCH-DMRS, from 1 subframe of data, for 3 MHz bandwidth.

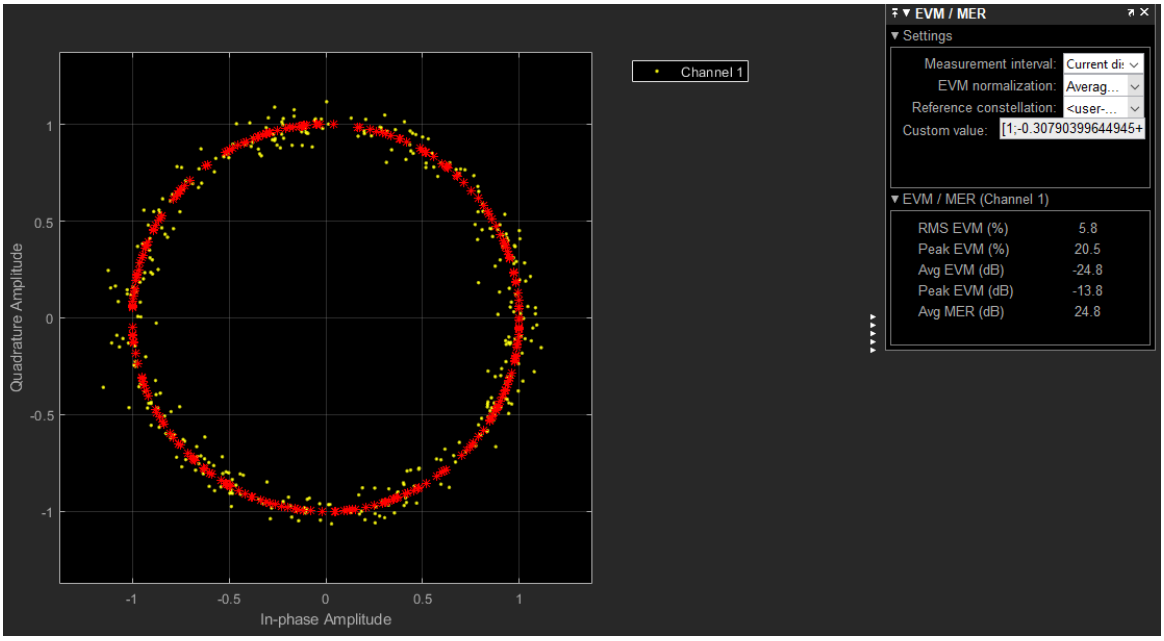

Figure B.8: Constellation diagram of PUSCH-DMRS with added noise, from 1 subframe of data, for 3 MHz bandwidth.

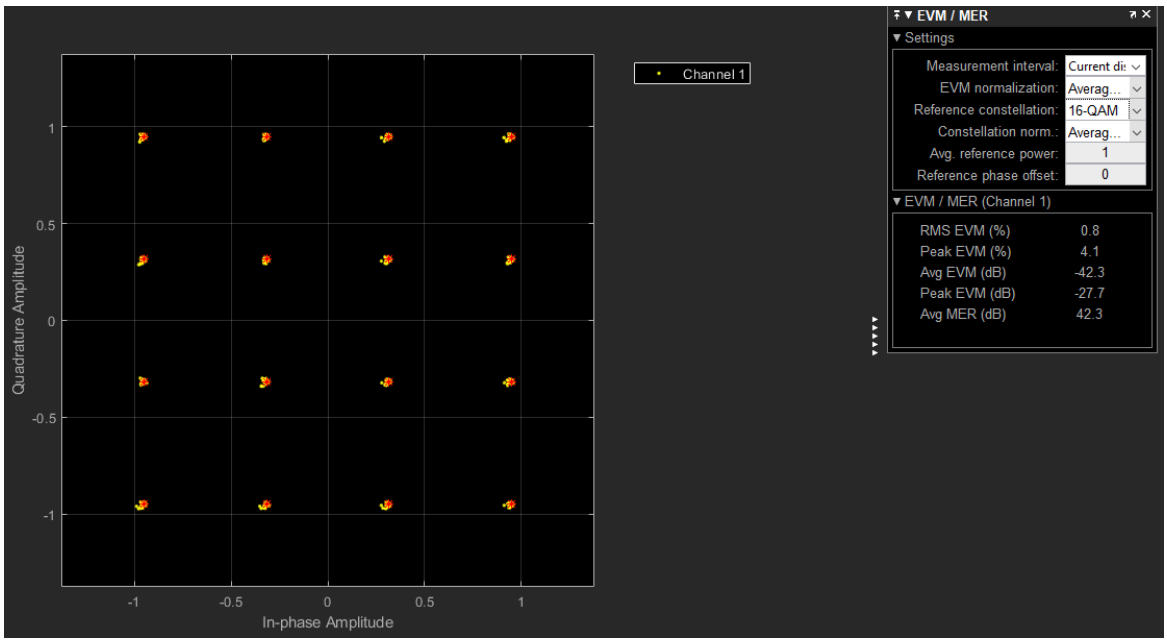

Figure B.9: Constellation diagram of PUSCH, from 1 subframe of data, for 5 MHz bandwidth.

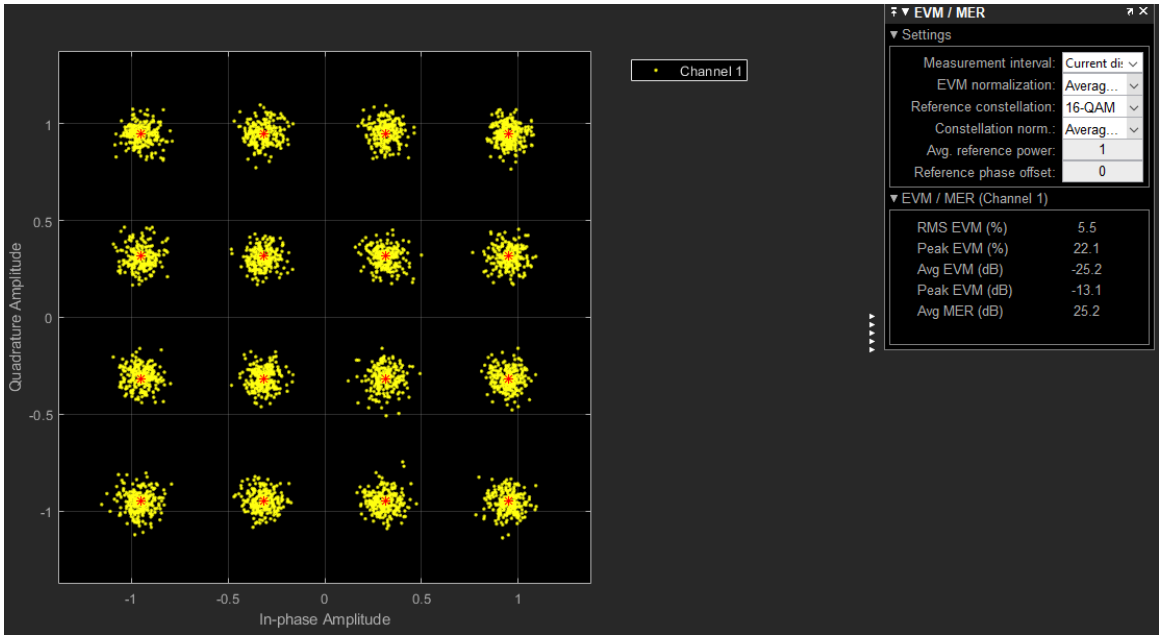

Figure B.10: Constellation diagram of PUSCH with added noise, from 1 subframe of data, for 5 MHz bandwidth.

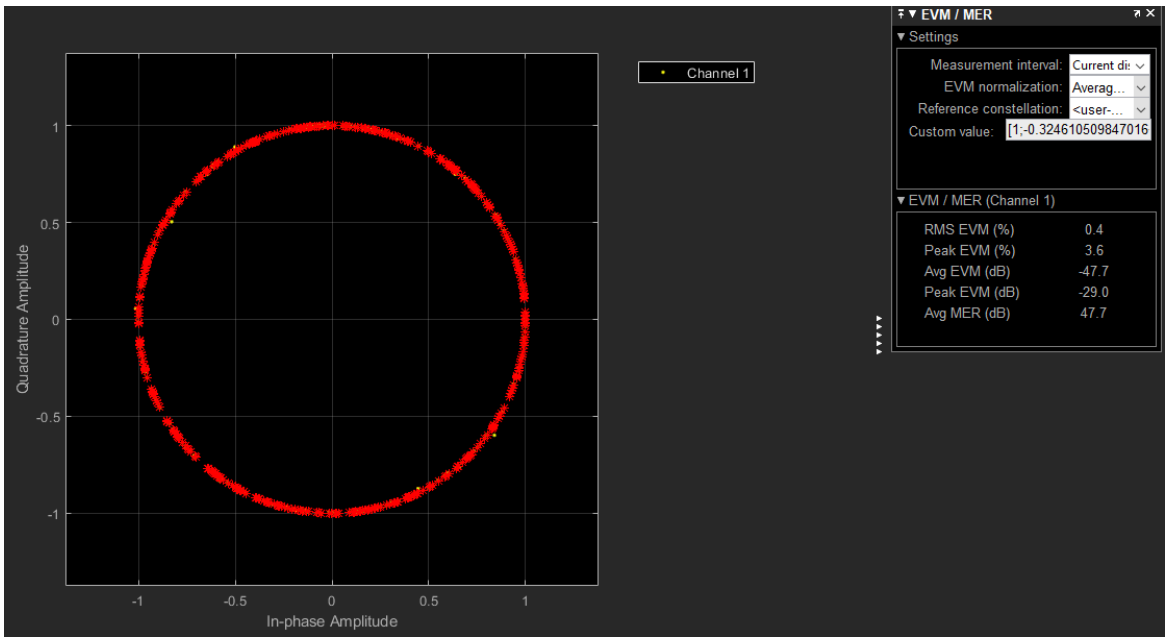

Figure B.11: Constellation diagram of PUSCH-DMRS, from 1 subframe of data, for 5 MHz bandwidth.

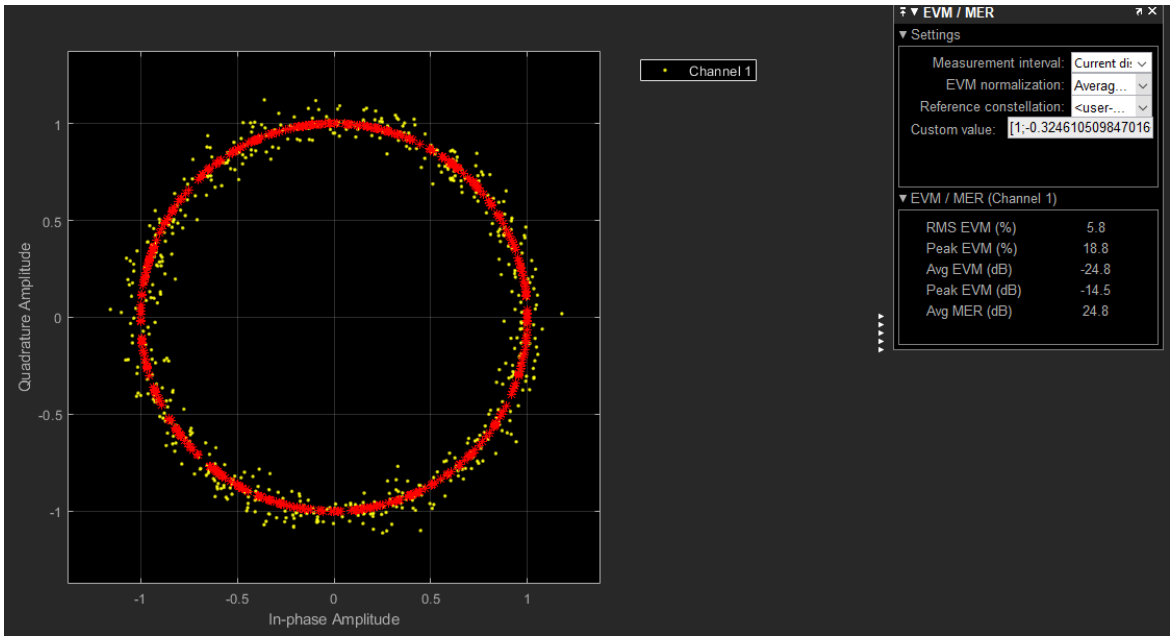

Figure B.12: Constellation diagram of PUSCH-DMRS with added noise, from 1 subframe of data, for 5 MHz bandwidth.

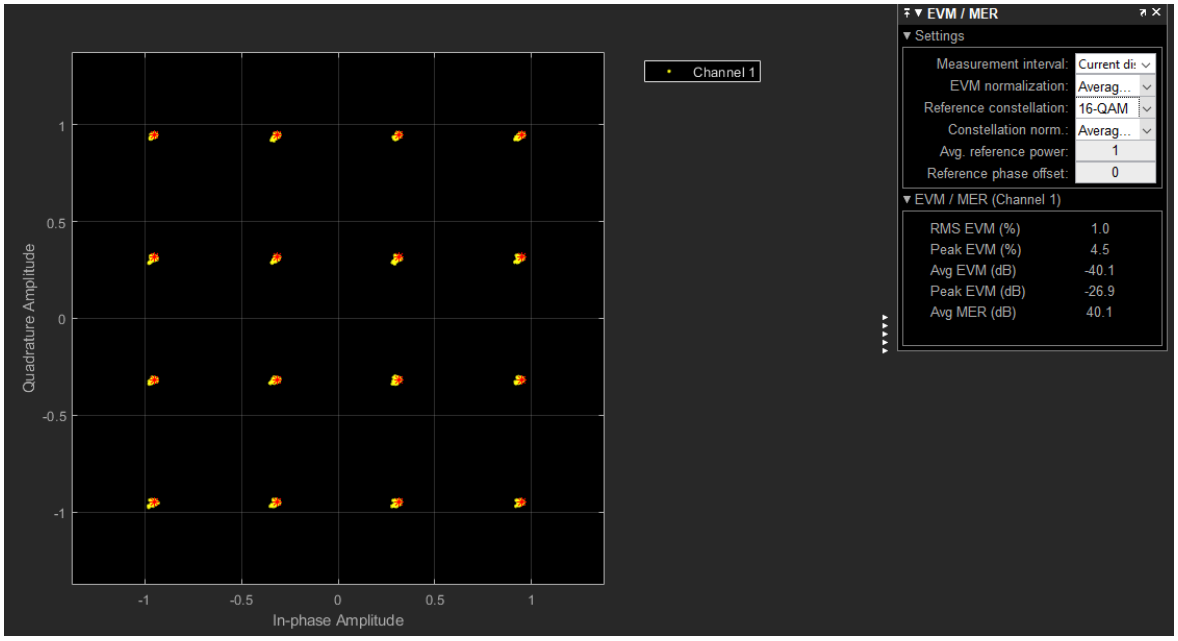

Figure B.13: Constellation diagram of PUSCH, from 1 subframe of data, for 10 MHz bandwidth.

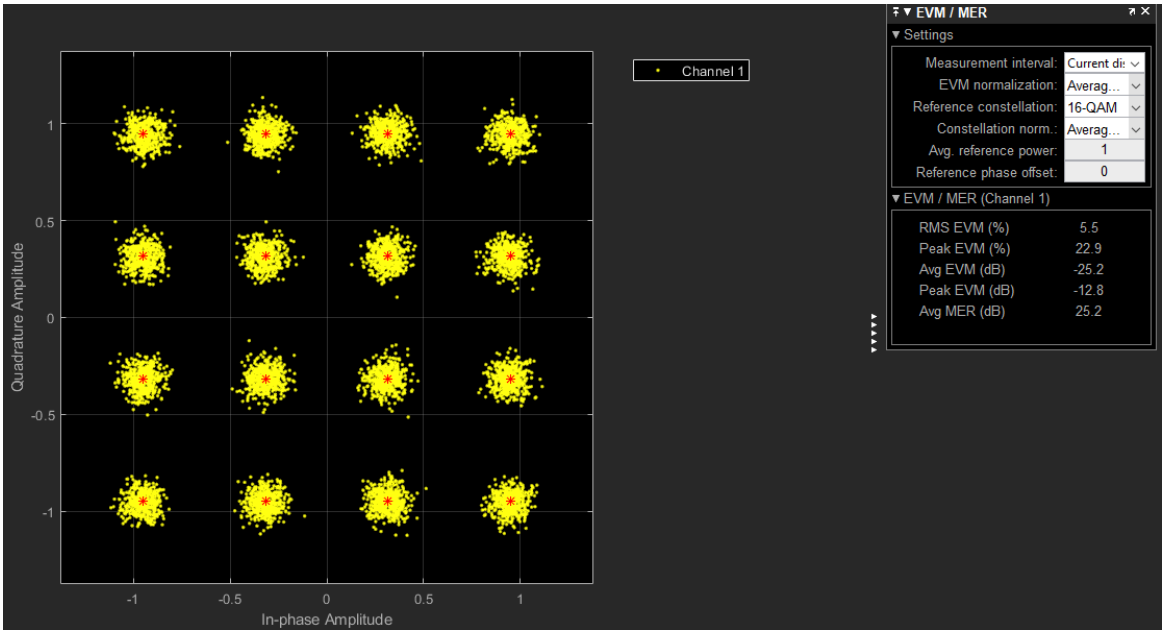

Figure B.14: Constellation diagram of PUSCH with added noise, from 1 subframe of data, for 10 MHz bandwidth.

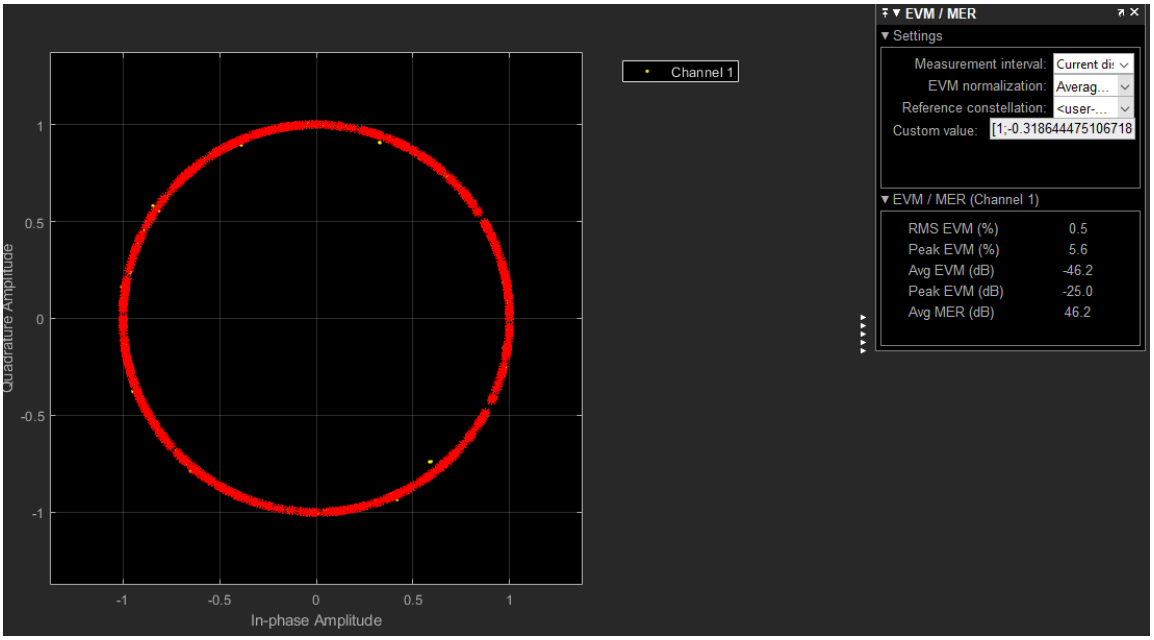

Figure B.15: Constellation diagram of PUSCH-DMRS, from 1 subframe of data, for 10 MHz bandwidth.

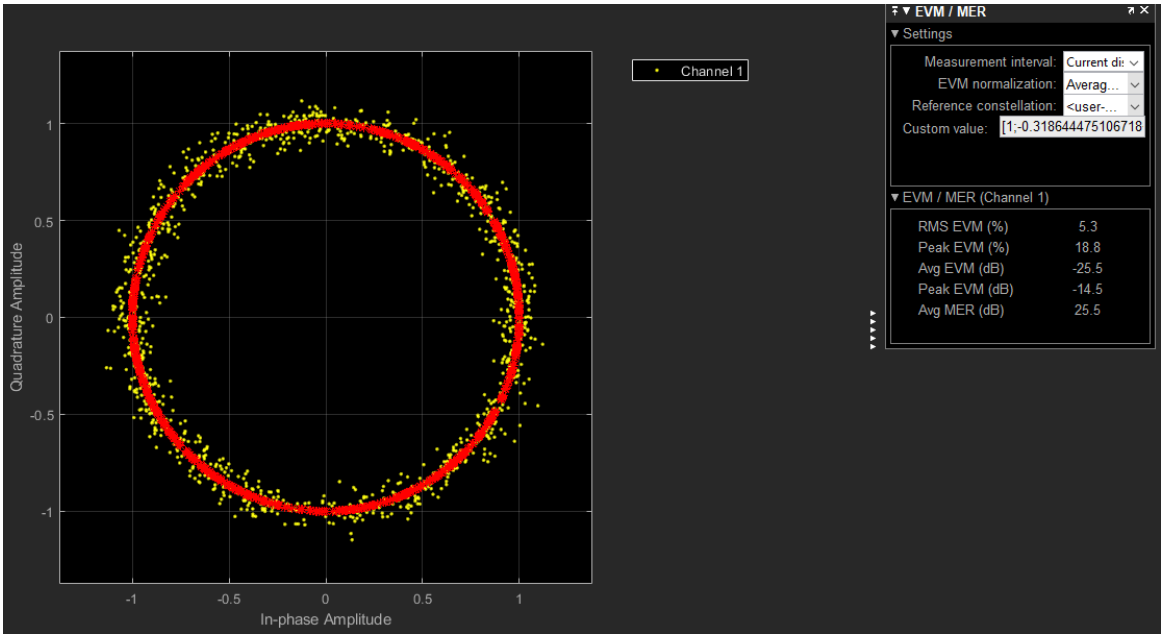

Figure B.16: Constellation diagram of PUSCH-DMRS with added noise, from 1 subframe of data, for 10 MHz bandwidth.

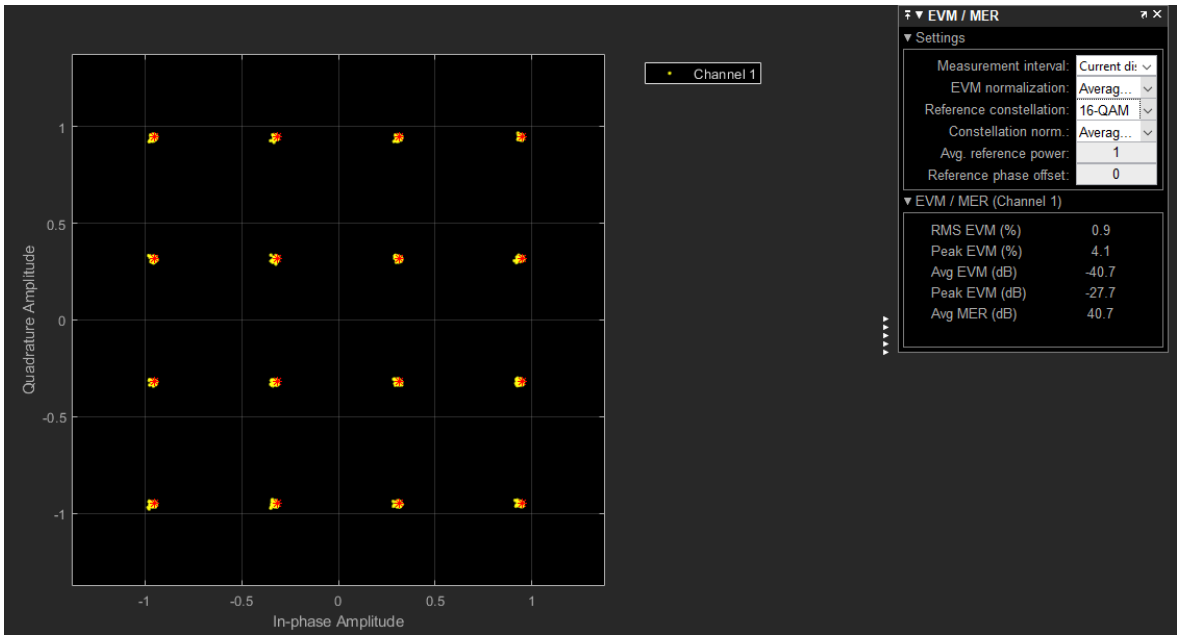

Figure B.17: Constellation diagram of PUSCH, from 1 subframe of data, for 15 MHz bandwidth.

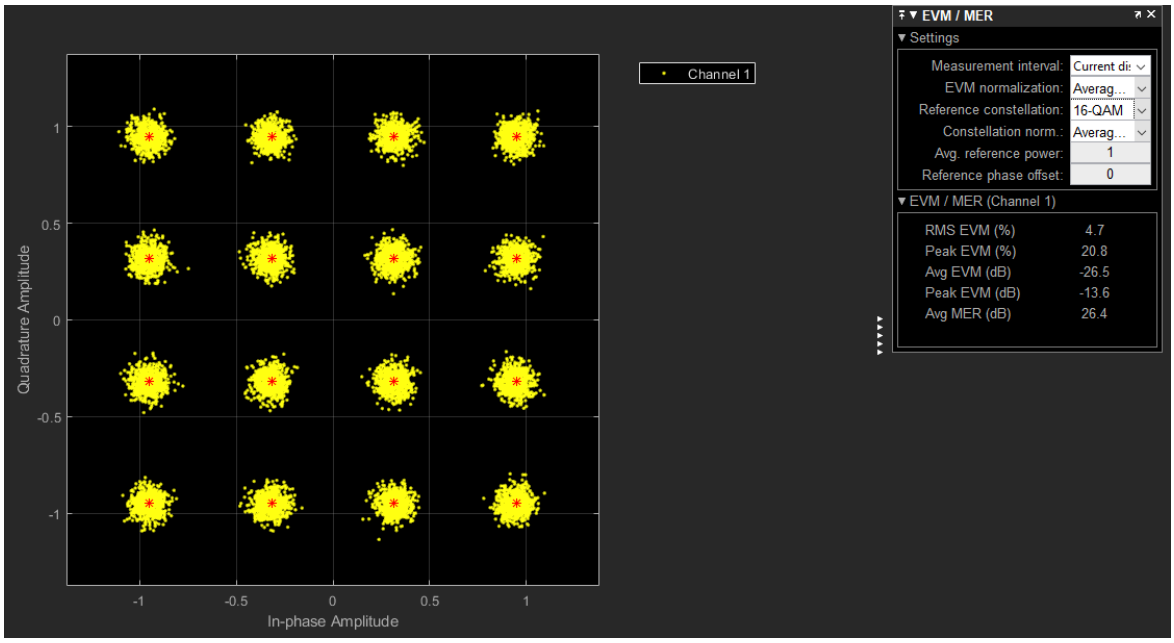

Figure B.18: Constellation diagram of PUSCH with added noise, from 1 subframe of data, for 15 MHz bandwidth.

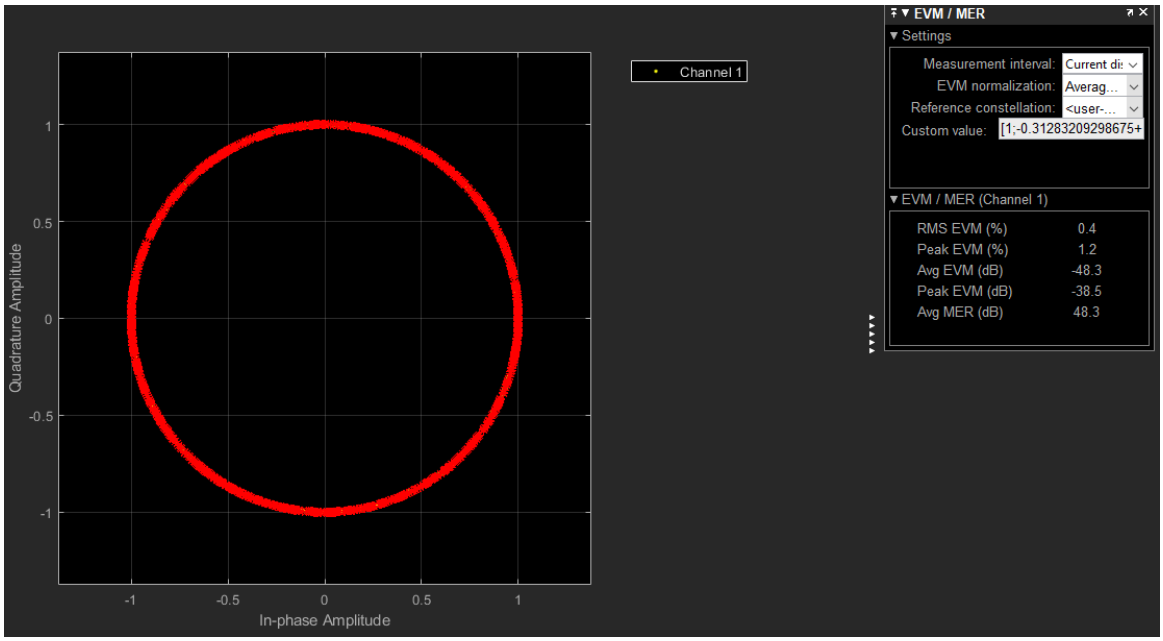

Figure B.19: Constellation diagram of PUSCH-DMRS, from 1 subframe of data, for 15 MHz bandwidth.

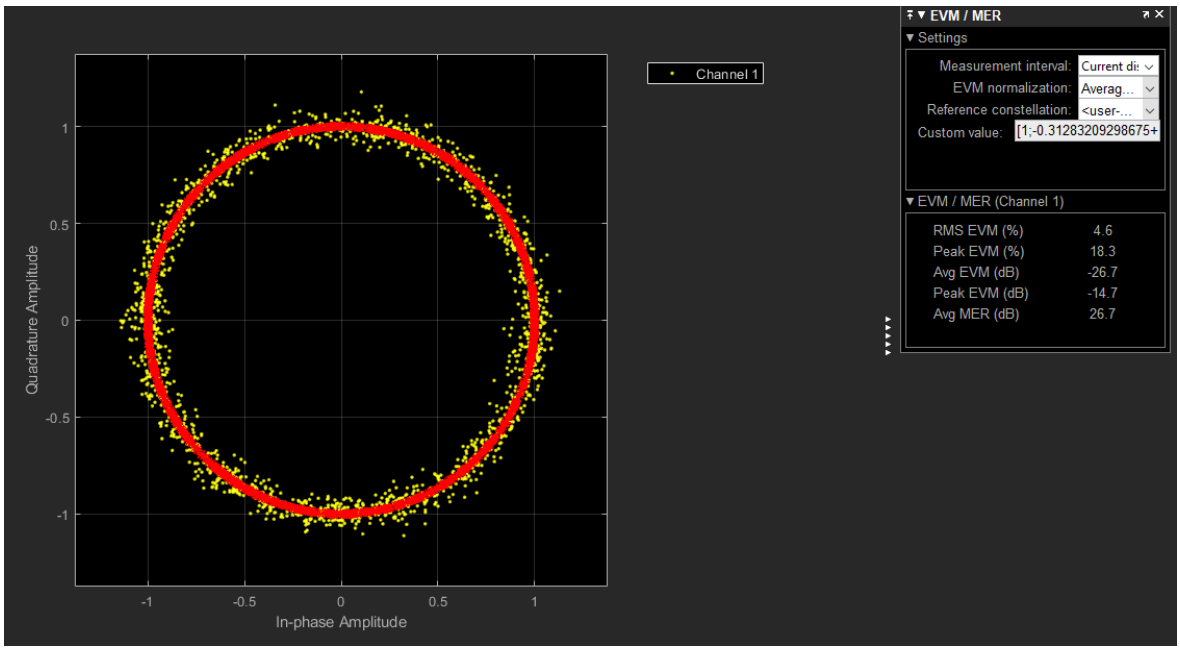

Figure B.20: Constellation diagram of PUSCH-DMRS with added noise, from 1 subframe of data, for 15 MHz bandwidth.

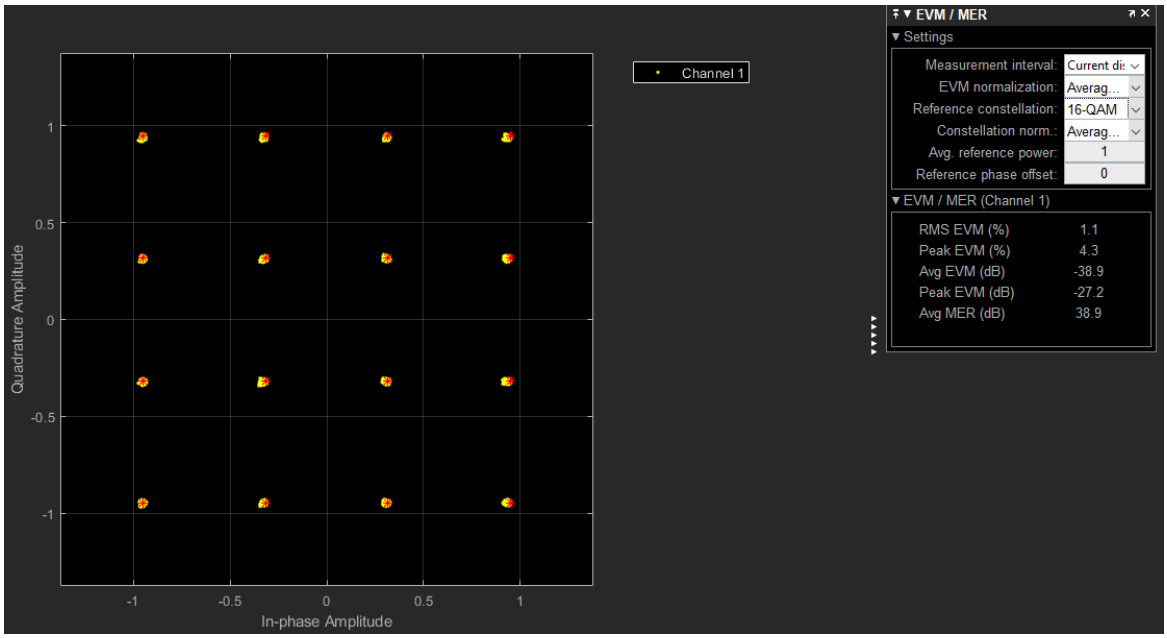

Figure B.21: Constellation diagram of PUSCH, from 1 subframe of data, for 20 MHz bandwidth.

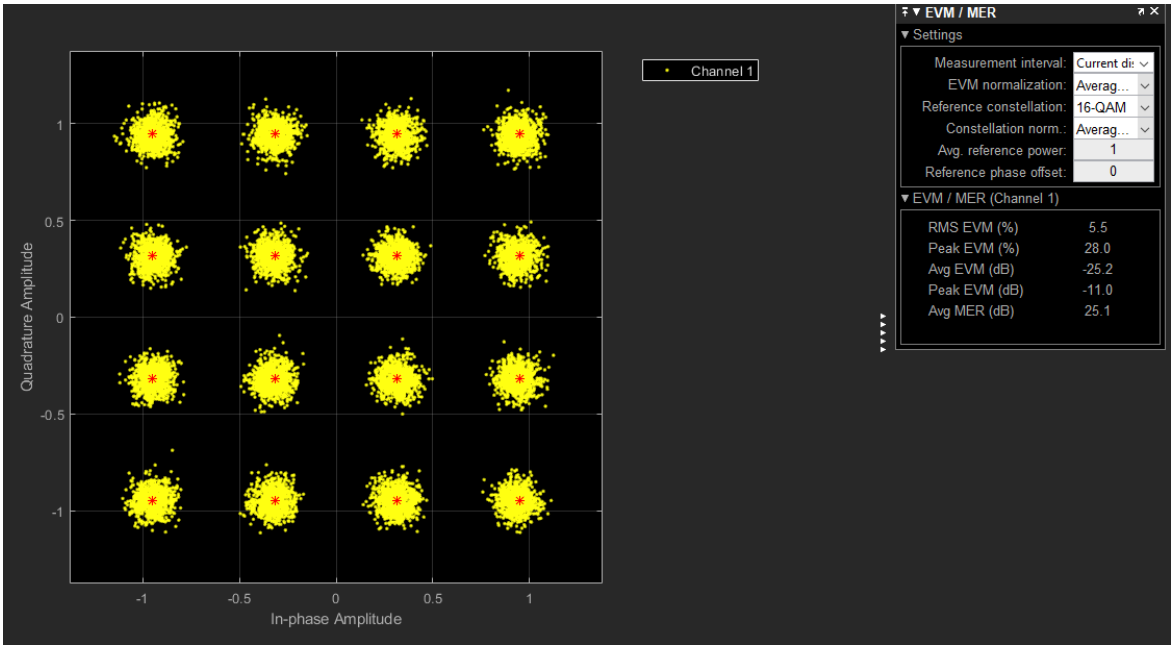

Figure B.22: Constellation diagram of PUSCH with added noise, from 1 subframe of data, for 20 MHz bandwidth.

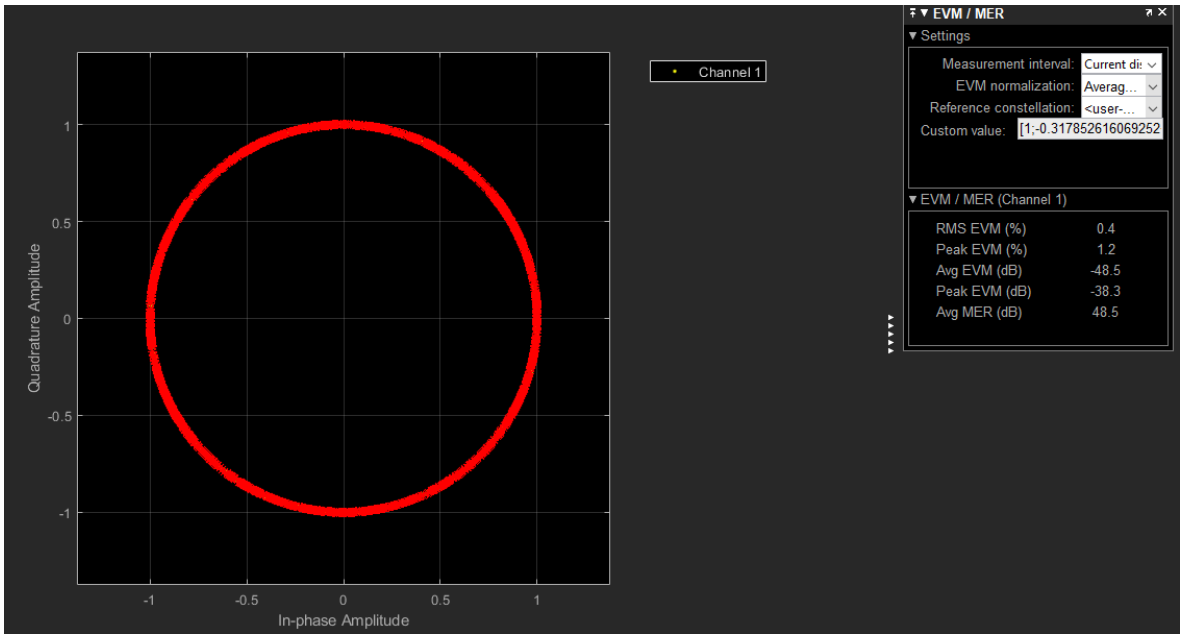

Figure B.23: Constellation diagram of PUSCH-DMRS, from 1 subframe of data, for 20 MHz bandwidth.

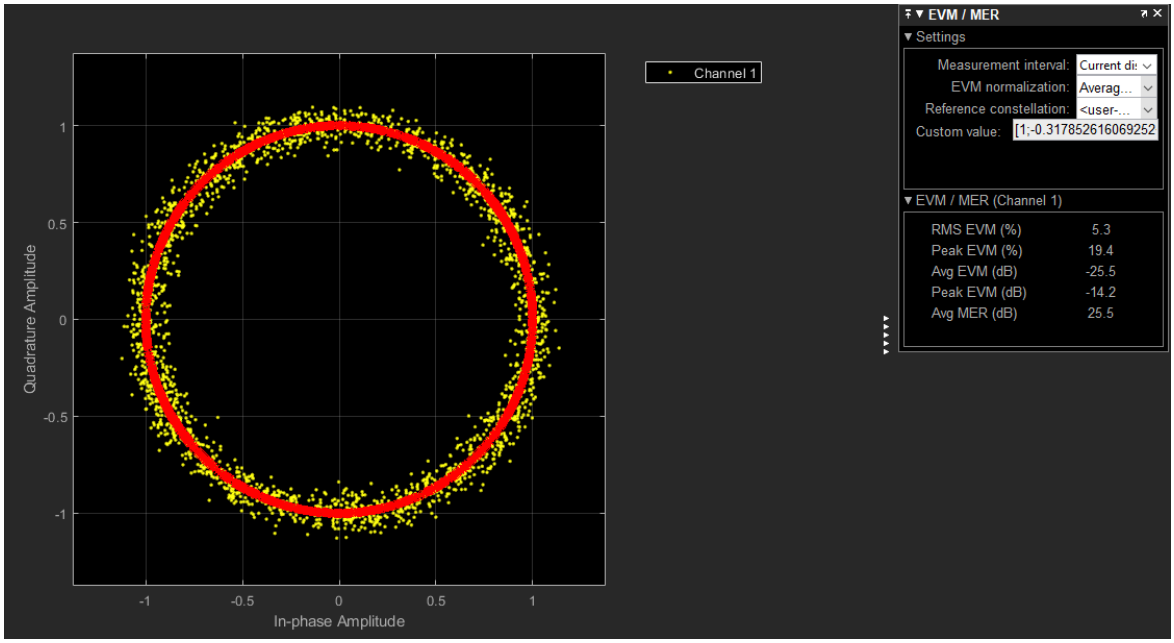

Figure B.24: Constellation diagram of PUSCH-DMRS with added noise, from 1 subframe of data, for 20 MHz bandwidth.