Universidade de Aveiro Departamento de Eletrónica,
Telecomunicações e Informática
2018

**Pedro Miguel
Costa Marques**

Fronthaul C-RAN baseado em Ethernet

Ethernet-based C-RAN Fronthaul

Universidade de Aveiro
2018

Departamento de Eletrónica,
Telecomunicações e Informática

Pedro Miguel
Costa Marques

Fronthaul C-RAN baseado em Ethernet

Ethernet-based C-RAN Fronthaul

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor Arnaldo Oliveira, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Paulo Monteiro, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president	Professor Doutor Armando Carlos Domingues da Rocha Professor Auxiliar da Universidade de Aveiro
vogais / examiners committee	Professora Doutora Maria do Carmo Raposo de Medeiros Professora Associada da Universidade de Coimbra (Arguente)
	Professor Doutor Arnaldo Silva Rodrigues de Oliveira Professor Auxiliar da Universidade de Aveiro (Orientador)

agradecimentos / acknowledgements

Deixo profundo agradecimento aos meus pais pelo suporte ao longo deste percurso académico.

Aos meus amigos e colegas que fizeram parte desse mesmo percurso.

Agradecimento especial ao Jorge Santos por toda a ajuda e ao meu orientador, Professor Doutor Arnaldo Silva Rodrigues de Oliveira, pela oportunidade e confiança depositada para fazer parte deste projecto.

Por fim, ao Instituto de Telecomunicações de Aveiro pelo suporte e infraestrutura.

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)].

Cofinanciado por:

Lisb@2020 COMPETE 2020 PORTUGAL 2020 UNIÃO EUROPEIA
Fundo Europeu
de Desenvolvimento Regional

Palavras Chave C-RAN, Fronthaul, FPGA, Ethernet, CPRI, RRH, BBU, 5G

Resumo

Durante a última década o tráfego de dados móveis tem aumentado a um ritmo impressionante. A proliferação de dispositivos móveis juntamente com serviços consumidores de grande largura de banda como *streaming* de vídeo e música, redes sociais e serviços na *cloud* têm colocado grande pressão na infraestrutura da rede móvel. Para suportar este aumento massivo de utilizadores e largura de banda a próxima geração de telecomunicações móveis – o 5G – explora novos conceitos, entre eles a utilização de bandas de frequências mais elevadas e a massificação das estações base. A este tipo de requisitos junta-se o facto da ineficiência da co-localização do processamento junto da unidade de rádio que incentiva a uma restruturação da arquitectura tradicional das redes móveis. Neste cenário surge o paradigma C-RAN, que pretende centralizar todo o processamento em banda base (BBU) e substituir as *base stations* atuais por soluções mais simples, eficientes e compactas que englobam apenas o processamento da parte de rádio e respetivo *front-end* de rádio frequência (RRH). Para além destes beneficios, a centralização do processamento facilita a virtualização e partilha de recursos, a gestão da interferência e tecnologias de processamento cooperativo. Esta divisão de funções traz no entanto alguns desafios no que diz respeito a largura de banda, taxas de dados e latências na interligação entre BBUs e RRHs – o *fronthaul*. Standards atualmente utilizados no link de *fronthaul* como o CPRI não foram originalmente desenhados para aplicações desta dimensão e apresentam algumas limitações, sendo intrinsecamente pouco flexíveis e eficientes. Acredita-se que outro tipo de abordagem, baseada em comutação de pacotes, poderia mitigar alguns destes problemas para além de trazer vantagens como a multiplexagem estatística, routing flexível e compatibilidade com redes de comutação de pacotes actuais. Apresentam no entanto vários desafios a nível de latência e sincronização associados.

Este trabalho de dissertação foca-se então no estudo e desenvolvimento de uma solução para o *fronthaul* baseada em *10 Gigabit Ethernet* sobre fibra ótica. O desenvolvimento será feito em dois *kits* de desenvolvimento baseados em Field Programmable Gate Array (FPGA) e implementado num demonstrador C-RAN já operacional - com *fronthaul* atualmente baseado em CPRI - no Instituto de Telecomunicações de Aveiro.

Keywords C-RAN, Fronthaul, FPGA, Ethernet, CPRI, RRH, BBU, 5G

Abstract

For the last decade mobile data traffic has been increasing at impressive rates. The proliferation of mobile devices together with high-bandwidth services like video and music streaming, social media and other cloud services have increased the load on top of the mobile network infrastructure. In order to support this massive increase in both users and bandwidth the next generation of mobile telecommunications network - 5G - explores new approaches, like the utilization of new frequency bands and the densification of base stations. This kind of requirements along with the inefficiency of the co-location of base band processing near the radio units encourages a rethink of traditional radio access networks. In this scenario emerges the C-RAN paradigm that intend to centralize all the base band processing (BBU) and replace current base stations for simpler, more efficient and compact solutions that only incorporate the radio front-end and respective radio processing (RRH). In addition to these benefits, centralized processing facilitates virtualization and resource sharing, interference management and cooperative processing technologies. This split of functions brings however, some challenges in respect to the data rates, bandwidth and latency in the link that connects BBUs and RRHs - the fronthaul. Today's existing standards like CPRI weren't originally designed for such applications and present some intrinsic bandwidth and flexibility limitations. It's considered that another approach, based on packet switching, could mitigate some of these problems in addition to bring some advantages such as statistical multiplexing, flexible routing and compatibility with current widespread packet switching networks. They do however, present a number of challenges regarding latency and synchronization.

This dissertation work focuses on the study and development of a fronthaul solution based in 10 Gigabit Ethernet over optical fiber. Development is done on top of two development kits based in Field Programmable Gate Array (FPGA) and implemented in an already operational C-RAN test-bed - currently with CPRI based fronthaul - at the Instituto de Telecomunicações - Aveiro.

Contents

List of Figures

List of Tables

Glossary

3GPP	3rd Generation Partnership Project
ADC	Analog-to-Digital Converter
AFE	Analog Front-End
AMBA	Advanced Microcontroller Bus Architecture
AWG	Arrayed Waveguide Grating
AXI	Advanced eXtensible Interface
BBU	Base Band Unit
BSC	Base Station Controller
BTS	Base Transceiver Station
CAPEX	Capital Expenditure
CDMA	Code Division Multiple Access
CN	Core Network
CoMP	Coordinated Multiple Point
COTS	Commercial Off-The-Shelf
CPRI	Common Public Radio Interface
C-RAN	Cloud/Centralized/Cooperative - Radio Access Network
DAC	Digital-to-Analog Converter
DMA	Direct Memory Access
DSP	Digital Signal Processor
DWDM	Dense Wavelength Division Multiplexing
eCPRI	enhanced CPRI
EDGE	Enhanced Data rates for GSM Evolution
eMBB	enhanced Mobile Broadband
eNB	evolved Node B
EPC	Evolved Packet Core
EUTRAN	Evolved Universal Terrestrial Radio Access Network
EVM	Error Vector Magnitude
FCS	Frame Check Sequence
FFT	Fast Fourier Transform
FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
GERAN	GSM/EDGE Radio Access Network
GMII	Gigabit Media-Independent Interface
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications

HARQ	Hybrid Automatic Retransmit reQuest
HDLC	High level Data Link Control
HPC	High Pin Count
HSPA	High Speed Packet Access
HSS	Home Subscriber Server
IFFT	Inverse Fast Fourier Transform
ILA	Integrated Logic Analyzer
IoT	Internet of Things
LPC	Low Pin Count
LTE	Long Term Evolution
LUT	Look Up Table
LVCMOS	Low Voltage Complementary Metal Oxide Semiconductor
LVDS	Low Voltage Differential Signaling
LVPECL	Low Voltage Positive Emitter-Coupled Logic
M2M	Machine-to-Machine
MAC	Medium Access Control
MCC	Mobile Country Code
MGT	Multi-Gigabit Transceiver
MIMO	Multiple Input Multiple Output
MMCM	Mixed-Mode Clock Manager
MME	Mobility Management Entity
MMF	Multi-Mode Fiber
mMTC	massive Machine Type Communications
MNC	Mobile Network Code
MNO	Mobile Network Operator
MPLS	Multi-Protocol Label Switching
NFV	Network Function Virtualization
OAI	Open Air Interface
OBSAI	Open Base Station Architecture Initiative
OPEX	Operational Expenditure
PCIe	Peripheral Component Interconnect express
PCS	Physical Coding Sublayer
P-GW	Packet Data Network Gateway
PLL	Phase Locked Loop
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
PON	Passive Optical Network
PTP	Precision Time Protocol
RAN	Radio Access Network
RAT	Radio Access Technology
RE	Radio Equipment
REC	Radio Equipment Controller
RF	Radio Frequency
RGMII	Reduced Gigabit Media-Independent Interface
RNC	Radio Network Controller
RRH	Remote Radio Head

SDK	Software Development Kit
SDN	Software Defined Networking
SDR	Software Defined Radio
SFD	Start of Frame Delimiter
SFP	Small Form-factor Pluggable
SGMII	Serial Gigabit Media-Independent Interface
S-GW	Serving Gateway
SMA	SubMiniature version A
SoC	System-on-a-Chip
SPI	Serial Peripheral Interface
SyncE	Synchronous Ethernet
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
URLLC	Ultra Reliable Low Latency Communications
USB	Universal Serial Bus
UTRAN	Universal Terrestrial Radio Access Network
VHDL	Very High Speed Integrated Circuit Hardware Description Language
WDM	Wavelength Division Multiplexing
XGMII	10 Gigabit Media Independent Interface

CHAPTER 1

Introduction

1.1 Scope

Long gone is the time where "not-so-smart" phones were used to make simple phone calls and trade short text messages. Today's mobile devices use a myriad of services such as web browsing, video and music streaming, social media and other high-bandwidth services. This ascending trend of mobile devices together with new cloud services in the last years have led to an explosive growth in mobile data traffic.

According to [Cis17] global mobile traffic data increased more than 60% in 2016. This translates to 7.2 exabytes per month by the end of the year. Furthermore while 4G connections represent just 26% of all mobile connections (below 2G's 41% and 3G's 33%) it dominates the data traffic share with almost 70% of the total traffic. Cisco forecasts predict an increase in mobile traffic with a compound annual growth rate (CAGR) of 47% from 2016 to 2021. By 2021 it is expected to reach almost 50 exabytes per month corresponding to a five-fold increase over 2017 (Figure 1.1).

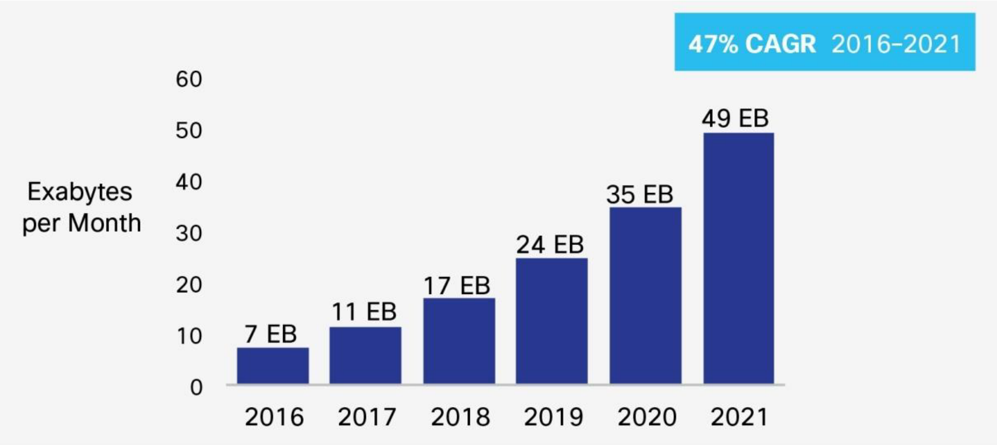


Figure 1.1: Predicted increase in monthly mobile data traffic from 2016 to 2021 (retrieved from [Cis17]).

These figures anticipate an largely increased number of devices and services contributing to an exponential traffic growth towards 5G [Lan+16]. Although the fifth generation of mobile communications still lack well defined specifications, there is a wide consensus that 5G networks must accommodate relative to 4G [Pen+15]:

- System capacity growth by a factor of 1000;
- Spectral efficiency by a factor of 10;
- Energy efficiency by a factor of 10;
- Data rate growth by a factor of 10.

5G is driven to enable three main use-cases illustrated in Figure 1.2: enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC).

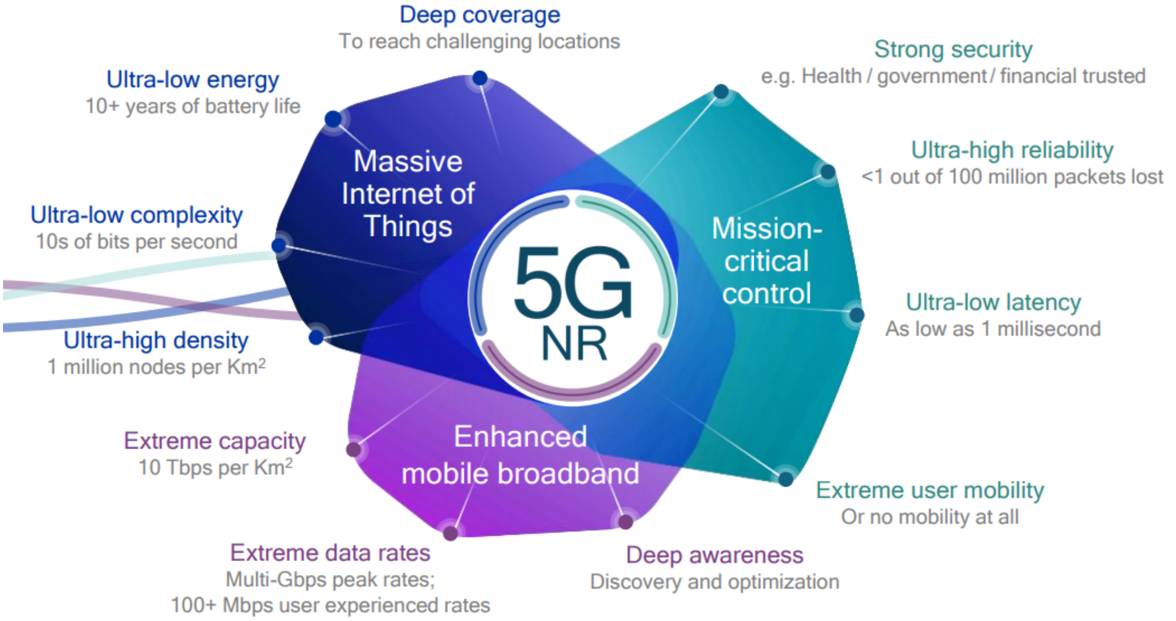


Figure 1.2: 5G use-cases (retrieved from [Qua16]).

This kind of requirements will demand more than current Radio Access Networks (RANs) have to offer and present a unique opportunity to rethink existing base station architectures in a sustainable way.

1.2 Motivation

In the search for a new, more efficient, high-bandwidth capable RANs emerges the Cloud/Centralized/Cooperative - Radio Access Network (C-RAN) paradigm. This new base station architecture operates with a simple underlying concept in mind: centralize the base band processing and share it with a large group of smaller, simpler and more efficient remote radio heads performing exclusively radio processing functions.

Traditional base station architectures perform all base-band processing on site and will be a limiting factor in next generation mobile telecommunications systems. Figure 1.3 illustrates how the current RANs could evolve in a phased manner. First by centralizing the base band processing units or digital unit (DU) with each DU dedicated to an exclusive radio unit (RU), followed by pooling the DUs together and cooperatively manage a group of RUs. Lastly obtaining the capability to virtualize the base band processing in a "DU cloud" where processing functions can be dynamically configured and allocated across different physical locations.

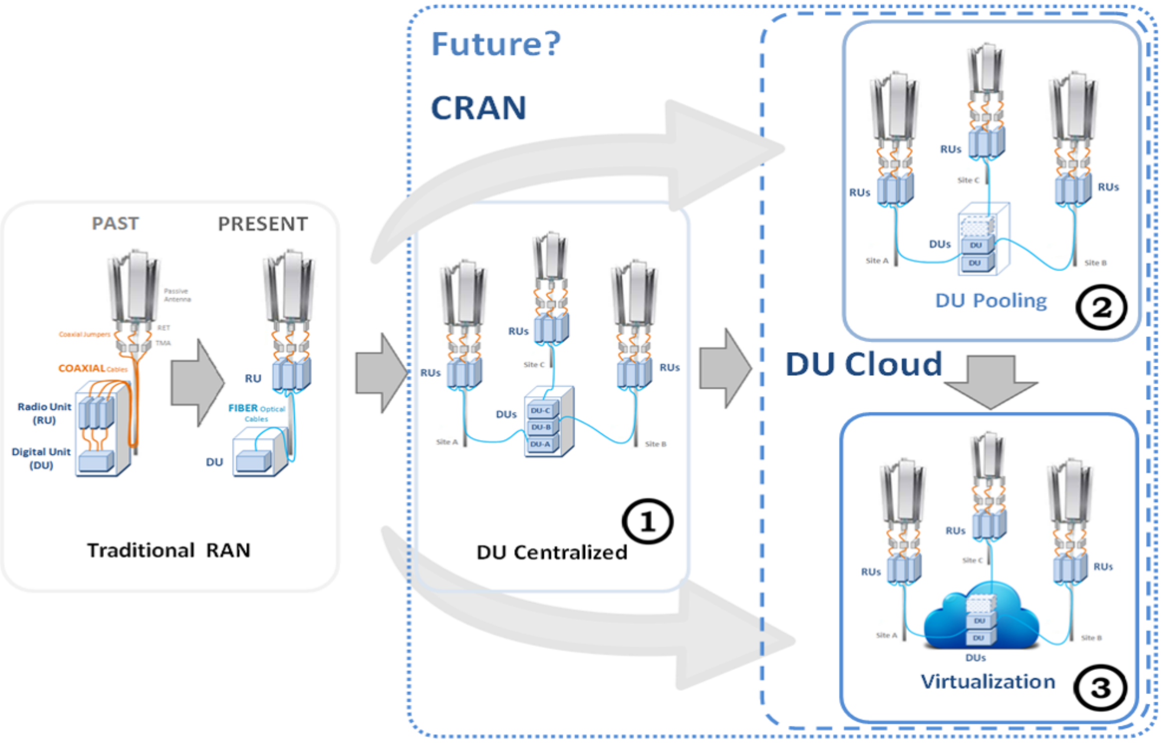


Figure 1.3: Evolution to C-RAN (retrieved from [Ora+]).

This paradigm would enable new technologies such as cloud computing, virtualization, Coordinated Multiple Point (CoMP), coordinated scheduling and coordinated beamforming. Connecting the baseband and radio elements is the fronthaul, a high-speed digital link transmitting information in the form of digitized IQ radio samples. There are however, challenges in terms of bandwidth and bit-rate. Current LTE-Advanced systems already support scalable carrier aggregation with bandwidths capable of reaching 100MHz. Assuming 16-bit samples and no compression, a single radio stream can easily reach 5 Gb/s [Gom+15]. Together with sites containing multiple Radio Access Technologies (RATs) and Multiple Input Multiple Output (MIMO) configurations, the aggregated bit-rate can easily hit tens or even hundreds of Gb/s [Pfe15].

Currently most used interface protocols in the fronthaul are CPRI and Open Base Station Architecture Initiative (OBSAI) with CPRI being the most widely adopted. Originally these transport standards were defined as an internal base station interface and have inherent bandwidth and flexibility limitations. It is considered that an Ethernet/packet based systems could mitigate the limitations and lead to a number of advantages over existing standards including statistical multiplexing and convergence with existing widespread packet-based network systems.

1.3 Objectives

The scope of this dissertation is then to develop a 10 Gigabit Ethernet based fronthaul solution and integrate it with Flexicell - an Open Air Interface (OAI) based C-RAN network with a 20km optical fronthaul currently deployed at Instituto de Telecomunicações - Aveiro [San+16b]. The framework is at present using the CPRI standard as fronthaul interface and it is the final objective to replace it with a fully functional 10Gbit Ethernet over fiber solution.

In order to achieve this objective, the following development stages were adopted:

1. Study and familiarization with current C-RAN framework.
2. Development of a standalone 10 Gigabit Ethernet link between two development boards based in FPGA.
3. Integration with the existing C-RAN test-bed.
4. Validation and testing.

1.4 Document Structure

The remainder of this document is organized as follows:

- **Chapter 2, Mobile Networks** - This chapter contextualizes the evolution of the mobile network generations and respective radio access networks that led to the upcoming 5G and associated C-RAN paradigm. RAN infrastructure, architectures and concepts are introduced.

- **Chapter 3, Fronthaul** - Dives into more detail in the fronthaul intricacies. Most commonly used protocols are briefly explained and alternatives presented.

- **Chapter 4, CPRI based C-RAN Test-bed** - Introduces the CPRI-based C-RAN laboratory test-bed: physical infrastructure, major logical blocks, and functionality. This is the underlying infrastructure which will serve as a base to the Ethernet based fronthaul implementation.

- **Chapter 5, 10 Gigabit Ethernet Link Implementation** - This chapter starts by describing the development stages of this work followed by the 10 Gb/s Ethernet link implementation in FPGA.

- **Chapter 6, Ethernet-based C-RAN Test-bed** - Integration of the new fronthaul implementation on top of the existent C-RAN laboratory test-bed.

- **Chapter 7, Validation, Optimization and Results** - Ethernet-based C-RAN testbed validation, optimization, tests and results.

- **Chapter 8, Conclusion and Future Work** - In this chapter conclusions are highlighted followed by proposed future lines of work.

- **Appendix A, Flexicell Start-Up User Guide** - Detailed user guide on how to successfully start-up the original CPRI-based laboratory C-RAN test-bed.

- **Appendix B, OpenAirInterface Deployment** - OpenAirInterface EPC and BBU (eNB) deployment.

CHAPTER 2

Mobile Networks

This chapter contextualizes the evolution of the mobile network generations and respective radio access networks that led to the upcoming 5G and associated C-RAN paradigm. RAN infrastructure, architectures and concepts are introduced.

2.1 Introduction

Since the first analog-based wireless telecommunications deployed back in the 80's to today's 4G networks, technology continue to evolve in order to deliver more and better services to the society, with each mobile generation emerging in the beginning of every decade so far.

The first generation (1G) started by providing basic voice calls based on analog signals. 2G introduced the digital era and the support for Short Message Service (SMS), Multimedia Messaging Service (MMS) and other simple data services. Besides that, the coverage and capacity were improved and the first digital standards like Global System for Mobile Communications (GSM) and Code Division Multiple Access (CDMA) based systems were created. Later modifications led to General Packet Radio Service (GPRS) and Enhanced Data rates for GSM Evolution (EDGE) networks (commonly referred to as 2.5G) allowed for higher data rates. 3G finally provided true mobile broadband communications with the proliferation of internet and multimedia applications to the mobile subscribers. Firstly with Universal Mobile Telecommunications System (UMTS) and CDMA-2000/EV-DO and later High Speed Packet Access (HSPA) (3.5G). In the past decade 4G arrived with the Long Term Evolution (LTE)/LTE-Advanced IP based standards and elevated all 3G offers to the next level. Technologies like MIMO, carrier aggregation and wider channel bandwidths delivered higher than ever data transmissions rates and lower latencies.

Nowadays the massive increase in mobile data traffic, the dawn of the Internet of Things (IoT) and the proliferation of sensors networks and Machine-to-Machine (M2M) communications is leading to the next, (fifth) generation of mobile communications (5G).

Although there is not yet a definitive set of features and specifications, technology research and development for 5G is already under way with the industry and academics working in the respective standardization and study of the most advantageous technologies capable of delivering the main imposed features: increased bandwidth, peak and average data rates; reduced latency and energy consumption; enhanced spectral efficiency and support for a huge amount of connected radio devices. Network availability is expected by 2020.

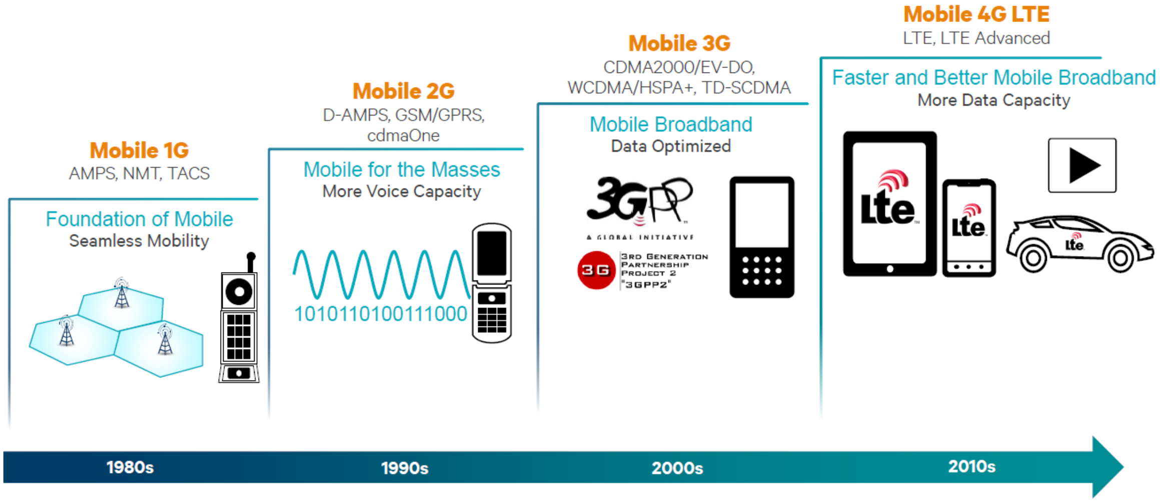


Figure 2.1: Evolution of the mobile generations and associated RATs (retrieved from [Qua14]).

Generally speaking, mobile telecommunication networks are composed by 3 main segments: the User Equipment (UE), the Radio Access Network (RAN) and the Core Network (CN) (Figure 2.2). The UEs are the mobile subscribers devices such as smartphones, tablets and laptops. The CN is, as the name implies, the central part of the network. It is responsible for the network management, maintaining the subscribers database, authentication, serve as an internet gateway, among others. Connecting the latter elements is the RAN, that is going to be the main focus of this dissertation and object of more detail in the following chapters.

2.2 Radio Access Networks (RANs)

Radio access networks are the edge element in a mobile telecommunications network that connects the user equipment to the core network by implementing the Radio Access Technology (RAT) that is ultimately going to be the medium used to exchange information through the air interface with mobile devices. Each release of a new generation is associated with new RATs that typically provide new functionality, improve coverage/capacity or simple boost the quality of the service and data rates.

Most employed RANs today are deployed in a distributed architecture. Each base station is connected by a backhaul link to the core network, usually making use of IP/Multi-Protocol Label Switching (MPLS) connections capable of reaching hundreds of kilometers while the antennas are directly connected to the base stations over short copper or optical fiber links.

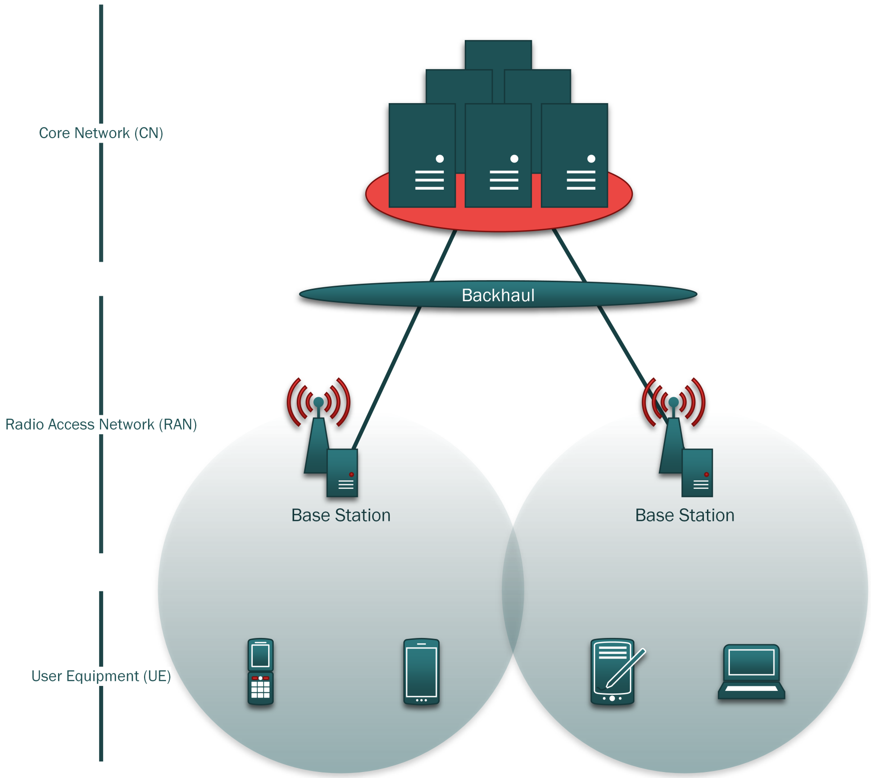


Figure 2.2: Generic distributed architecture of a mobile network.

To ensure proper coverage, different types of radio sites are deployed accordingly to the geographical area, forming a heterogeneous network. Macro-cells are used in suburban areas and are design to cover a radius of several km's and serve hundreds or thousands of users. Antennas for this bigger type of cells are mounted on ground-based masts, rooftops and other existing structures in order to ensure good coverage and can have a maximum output power of 100W [Fre13]. Depending on the topography of the area, there might be the need to complement the macro-cell coverage with so called small-cells. They can be micro, pico or femto-cells, with respective decreasing coverage radius and capacity. This type of cells provide cell coverage to a smaller, more focused area.

2.2.1 Evolution of the RAN

Before discussing further concepts it is essential to introduce how current RAN architectures work and why they need to evolve. Current radio access networks include second generation GSM/EDGE Radio Access Network (GERAN), third generation Universal Terrestrial Radio Access Network (UTRAN), and fourth generation Evolved Universal Terrestrial Radio Access Network (EUTRAN). The GERAN is based on GSM/EDGE radio access technologies and it is comprised by the Base Transceiver Station (BTS) that includes the antennas and Radio Frequency (RF) front-end to communicate with 2G mobile devices and takes care of the baseband processing relative to its users. Controlling a number of BTS is the Base Station Controller (BSC). They manage the radio access network and interface with the core network.

The UTRAN implements the UMTS RAT and although its elements maintain the same core function, they earn new terminology to differentiate from the older generation. The BTS evolve into Node B and BSC into the Radio Network Controller (RNC). Both GERAN and UTRAN use the same core network based in both circuit switching and packet switching.

Finally the EUTRAN introduce the LTE all IP standard along with some modifications to previous networks. Old generation RNC functions are moved to the now called evolved Node B (eNB) that links directly to the core network through the S1 interface. Different eNBs are now interconnected with a new X2 interface. This drive occurs to decrease connection set-up and facilitate handovers and real time data sessions. The LTE core network is now fully packet switched with the new Evolved Packet Core (EPC). It can however, still connect with older generations CN with internal interfaces. Figure 2.3 expose the latter elements in a block diagram for easier visualization.

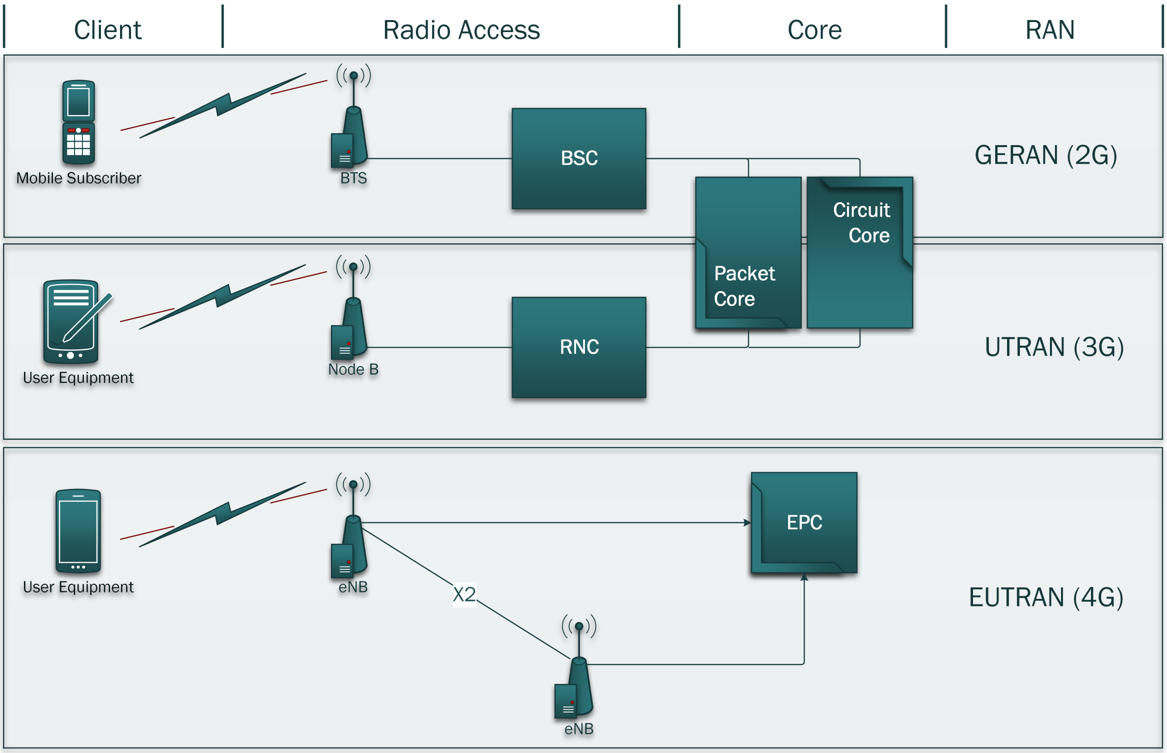


Figure 2.3: Block diagram of current RANs evolution, respective RATs and terminologies.

2.2.2 Limitations

Current radio access networks have been serving society for several years, but traditional architectures are starting to show some inherent drawbacks. As society demands wider coverage, better capacity and higher data rates, the Mobile Network Operators (MNOs) are forced to improve their network. In a traditional distributed architecture this can only be achieved by deploying more base stations and respective supporting equipment. This is far from being a simple task, as adding more base stations generally means that besides the cost of main hardware equipment (antennas, processing unit, transmission equipment, etc.) there are also extra costs associated with site acquisition (purchase or rental), civil works, network planning and auxiliary equipment like power and air conditioning to cool the processing units. Combined with fundamental flaws detailed in the next subsections, this leads to a rise of both Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) and greatly increase the Total Cost of Ownership (TCO). As shown in Figure 2.4, only 50% of the initial investment is translated in actual base station hardware/software plus transmission equipment. Same conclusions applies to the OPEX, with the largest chunks being devoted to site rental and electricity.

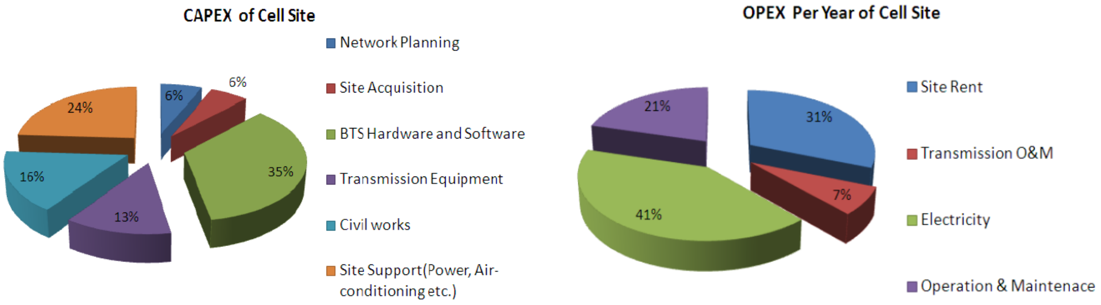


Figure 2.4: CAPEX/OPEX of a cell site (retrieved from [Ins13])

2.2.2.1 Power Consumption and Low Power Efficiency

Since in a traditional base station all its base processing is performed on site, a lot of processing power is needed and therefore so is the power to cool down the processing equipment. This translates to big cabinets at the bottom of the radio towers containing air conditioning units and multiple negative effects are derived from here. From one side, the need for local processing equipment and associated cooling solution derives in a large footprint and increases costs relative to civil works and site acquisition. On the other side, almost half of the total power consumption at the base station is allocated to the air conditioning equipment, resulting in low energy efficiency.

2.2.2.2 Poor Resource Allocation and Dynamic Load

As the name implies, a mobile network is highly dynamic by nature. As mobile subscribers move from residential districts to central office / commercial areas by morning and vice-versa by the end of day, a mobile traffic pattern is shaped - the tidal wave effect (Figure 2.5). This causes oversubscribed base stations in some areas and idling base stations in other areas.

Data collected from China Mobile cell sites (Figure 2.5) shows that there is an approximately 6 hour period between late night and early morning in which the network load is almost zero. Since mobile network operators need to ensure uninterrupted service for 24/7, base stations are designed to run at peak load with little to no energy savings compared to idle times. This leads to a considerable amount of processing capacity and power consumption being wasted [Ins13].

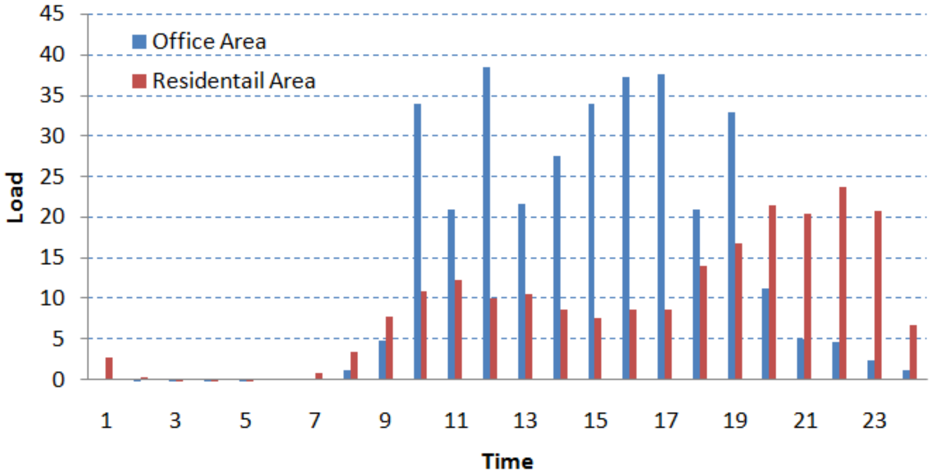


Figure 2.5: Tidal wave effect representing network load over a 24h period (retrieved from [Ins13]).

2.3 C-RAN Architecture

The C-RAN architecture aims to provide significant performance gains for next generation services while keeping the cost per bit low and spectral and energy efficiency high by restructuring the typical access networks. A C-RAN architecture decouples traditional base stations in two separated entities, centralizing the baseband processing in "resource pools" while cost-effective, power-efficient and small-footprint radio heads provides seamless coverage to the UEs.

To be noted that the C-RAN architecture is not a replacement for today 3G and 4G standards, only an alternative approach to current delivery and a promising RAN architecture capable of enabling 5G [I+15] [Ins13].

Typical C-RAN architecture consists of 3 major elements: the Base Band Unit (BBU), the RRH and the fronthaul link connecting the two (Figure 2.6).

2.3.1 BBU

The BBU pool is where the centralized baseband processing is done and consists in sets of digital baseband processing hardware/software that operate as virtualized base stations providing large scale processing and management of control signals from associated RRHs. The processing resources can then be dynamically allocated and reconfigured based on current traffic. The backhaul interface with the core network is also implemented here. [Pfe15] classifies them into three variants: A "BBU hotel" if each BBU in the cluster is dedicated to an exclusive RRH. A "BBU pool" where multiple collocated base band units collaboratively manage a group of RRHs. This is the preferred definition used in the rest of this dissertation. Lastly, a "BBU cloud", when processing functions can be dynamically configured and allocated across different physical locations.

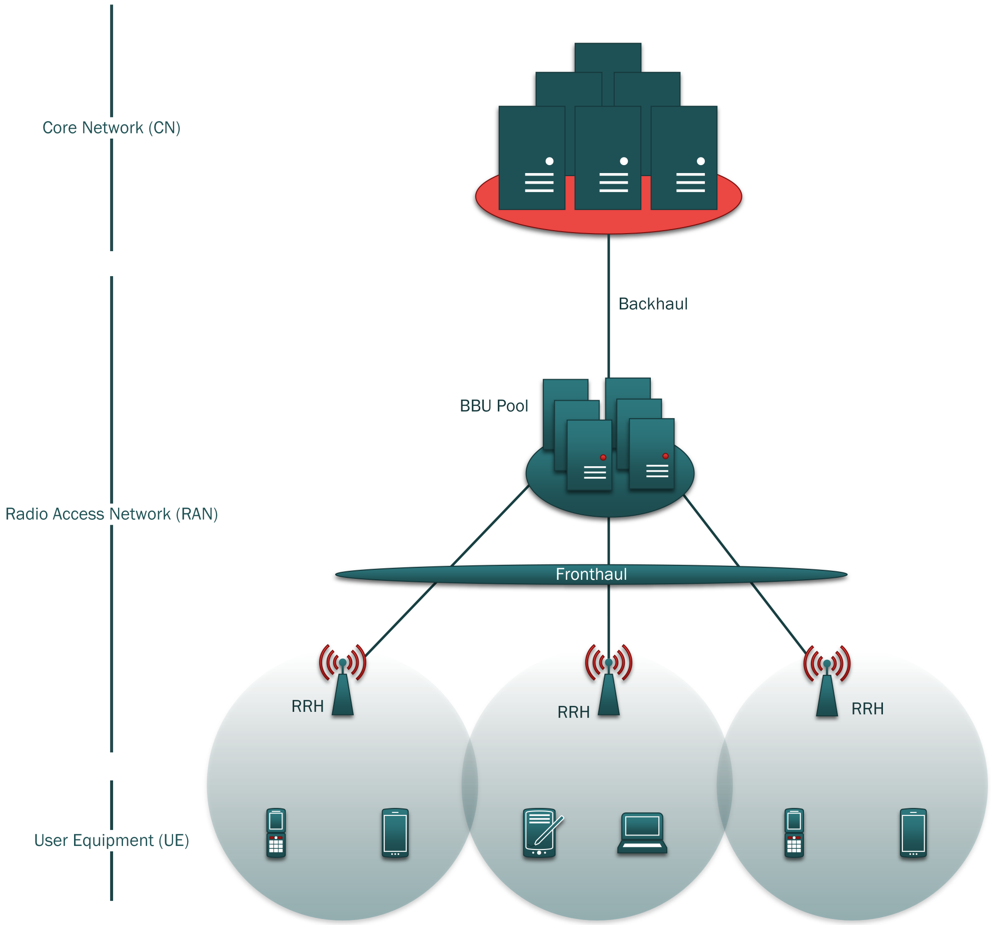


Figure 2.6: C-RAN architecture.

2.3.2 RRH

RRHs are small and power/cost efficient radio units located at the remote sites that ensure mobile network coverage. Since all or most of the base band signal processing is done at the BBU pool, the remote radio heads don't need much more hardware other than RF electronics thus being relatively small and simple. An RRH system can be divided into three main blocks. The analog front-end, digital front-end and a fronthaul interface. The analog front-end include: the antennas to transmit/receive wireless signals; a duplexer in case of a single antenna usage for both transmission and reception; power amplifiers (PA) to increase the signal power prior to transmission and low noise amplifiers (LNA) to amplify the received signal and lastly Analog-to-Digital Converters (ADCs) / Digital-to-Analog Converters (DACs) to convert signals between the analog and digital domain. Although some functions like up/down conversion and pre-distortion could be done in the analog domain, in recent years there's been a drive to move these functions to the digital domain with the use of Software Defined Radio (SDR). SDR systems replace traditional radio systems with fixed components by performing these functions digitally, usually based in a Field Programmable Gate Array (FPGA) or Digital Signal Processor (DSP) that can have its hardware/software easily reconfigurable. This allow SDR systems to support multiple radio standards and opposed to typical "fixed" radio solutions provide greater flexibility and a smaller footprint.

2.3.3 Fronthaul

Fronthaul is the new network segment linking RRHs and BBUs. Although copper solutions can be used, generally an optical based system is favoured as a result of higher data rates and bandwidths, lower latency and enabled interoperability with other networking systems.

Typical fronthaul interface protocols include CPRI and OBSAI with CPRI being the most widely used in the industry. Since the fronthaul will be the main subject of study in this dissertation, **Chapter 3, Fronthaul** will be fully devoted to studying it in more detail.

2.3.4 Function Splitting

Although in previous chapters the RAN has been assumed as a fully centralized solution, keeping some functions on the RRH side might prove beneficial for a number of reasons. As such, the C-RAN architecture can be either fully or partially centralized. A fully centralized solution would imply that the information traded between BBU and RRH would be in the form of IQ radio samples. Even with current standards like LTE and LTE Advanced this method would demand huge amounts of bandwidth. With 5G in the horizon the increased bandwidth required would be completely unfeasible. The same argument is valid to latency constrains. Therefore, a partially centralized RAN with some level of layer 1 / layer 2 processing done in the RRH while remaining higher layer functions move to the centralized BBU pool might be the best solution. This implies that some sort of compromise between fronthaul bandwidth/latency and RRH complexity is vital. More on this matter further ahead in the chapter.

2.3.5 Advantages

C-RAN architectures bring several advantages described in the following subsections.

2.3.5.1 CAPEX and OPEX reduction

C-RANs bring intrinsic CAPEX and OPEX benefits. Since the RRHs are much simpler and have a smaller footprint, costs can be cut down in site acquisition/rent, civil works, power and cooling systems. This bumps the overall system efficiency, reduce emissions and environmental impact and cut down the TCO for the mobile network operators.

Virtualized BBUs can be dynamically configured to support different RATs. Combined with open platform, general purpose processors and the recent rise of SDR solutions eases upgrading procedures analog-based multi-standard support [Ins13].

2.3.5.2 Scalability and Flexibility

With the need to improve both coverage and capacity, network expansion by MNOs turns out very straightforward as new remote radio units can be quickly deployed and connected to the existing centralized BBU pool. Antenna installation in small space sites also becomes feasible, promoting RRH densification. As the number of remote radio heads increases the network operator can incrementally upgrade the BBU hardware to accommodate additional processing power required. SDR enabled RRHs together with virtualized BBUs would provide great flexibility in supporting future radio access technologies.

2.3.5.3 Centralized/Cooperative/Cloud Processing

The centralized processing power decoupled from remote sites brings numerous advantages and enables a wide range of new technologies. In one hand with centralized processing it is much easier to implement advanced radio transmission techniques that improve coverage and throughput by reducing inter-cell interference and consequently boost spectral efficiency [Pfe15]. These techniques include:

- Coordinated Multiple Point (CoMP);
- Coordinated scheduling and beamforming;
- Joint transmission/reception.

On the other hand, rising software trends such as Software Defined Networking (SDN) and Network Function Virtualization (NFV) could also benefit greatly from short link latencies that centralized processing provides. In Figure 2.7 is depicted a C-RAN architecture combining these technologies. Multiple RRHs in a heterogeneous network are connected to the BBU pool through a fronthaul interface. Here the base band processing pool (designated in the figure as "Radio Cloud Center") is implemented at the top of several general purpose platforms, each governed by an Hypervisor instantiating and controlling several virtual BBUs concurrently.

Additionally, the tidal wave effect mentioned at 2.2.2.2 cease to be a problem as resources are dynamically allocated as needed. Even if RRHs load changes dynamically proportionately to the movement of mobile subscribers, virtual base stations can be selectively turned off to save energy. Furthermore, in case of partial failure in the BBU pool equipment, processing power can be shared with running BBUs to ensure continuous operation.

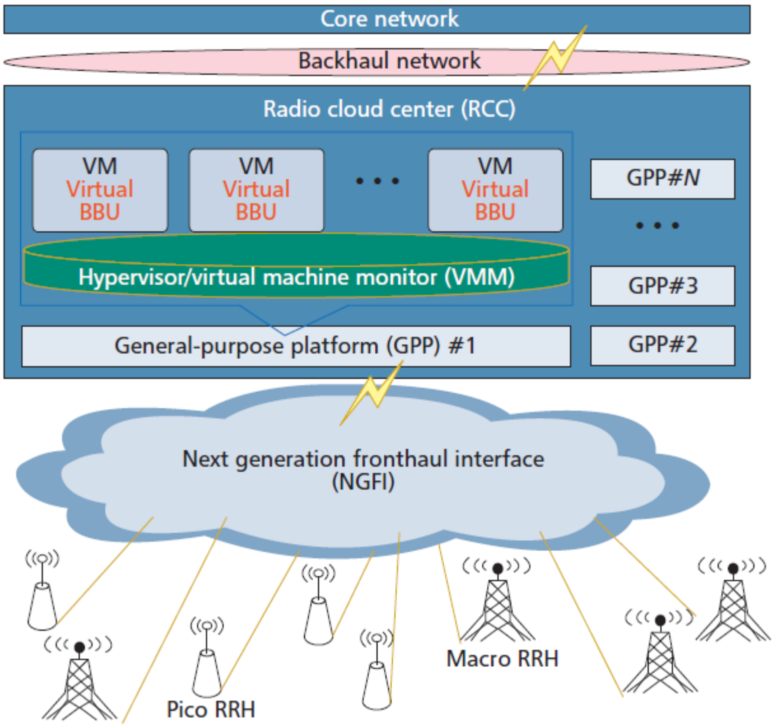


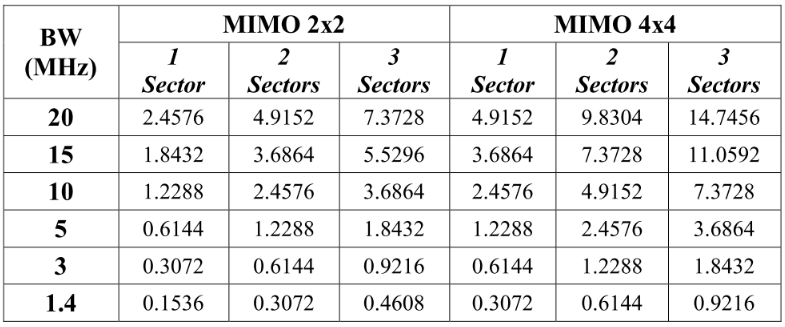
Figure 2.7: C-RAN architecture combining SDN and NFV (retrieved from [I+15]).

2.3.6 Challenges

Despite obvious multiple advantages, the C-RAN paradigm comes with it own challenges primarily in form of fronthaul constrains such as bandwidth, latency and synchronization.

2.3.6.1 Bandwidth

Current LTE-Advanced systems are already capable of supporting carrier aggregation with bandwidths reaching 100MHz. Assuming 16 bits per sample without compression, a single radio stream can easily reach 5 Gb/s [Gom+15]. Along with cell sites containing multiple radio access technologies and MIMO configurations, the aggregate bit rate can easily reach tens or even hundreds of Gb/s [Pfe15].



BW (MHz)	**MIMO 2x2**			**MIMO 4x4**		
	1 Sector	***2 Sectors***	***3 Sectors***	***1 Sector***	***2 Sectors***	***3 Sectors***
20	2.4576	4.9152	7.3728	4.9152	9.8304	14.7456
15	1.8432	3.6864	5.5296	3.6864	7.3728	11.0592
10	1.2288	2.4576	3.6864	2.4576	4.9152	7.3728
5	0.6144	1.2288	1.8432	1.2288	2.4576	3.6864
3	0.3072	0.6144	0.9216	0.6144	1.2288	1.8432
1.4	0.1536	0.3072	0.4608	0.3072	0.6144	0.9216

Table 2.1: Required CPRI line bit-rate for LTE in Gbps (retrieved from [Anj+15])

Table 2.1 shows how the required bit-rate scales for the transport of LTE IQ radio samples (over CPRI) with various bandwidths, MIMO configurations and number of sectors. In the most extreme scenario, with a bandwidth of 20MHz and a 4x4 MIMO configuration with 3 sectors, the required bit-rate reaches almost 15 Gb/s. With the expected increased bandwidth and Massive-MIMO configurations predicted for the 5G these numbers will grow exponentially.

2.3.6.2 Latency

Currently the maximum distance between RRH and BBU or in other words, the maximum fronthaul link length is constrained by the timing requirement of the Hybrid Automatic Retransmit reQuest (HARQ) protocol used as a retransmission mechanism between UE and eNB in an LTE network [Har14].

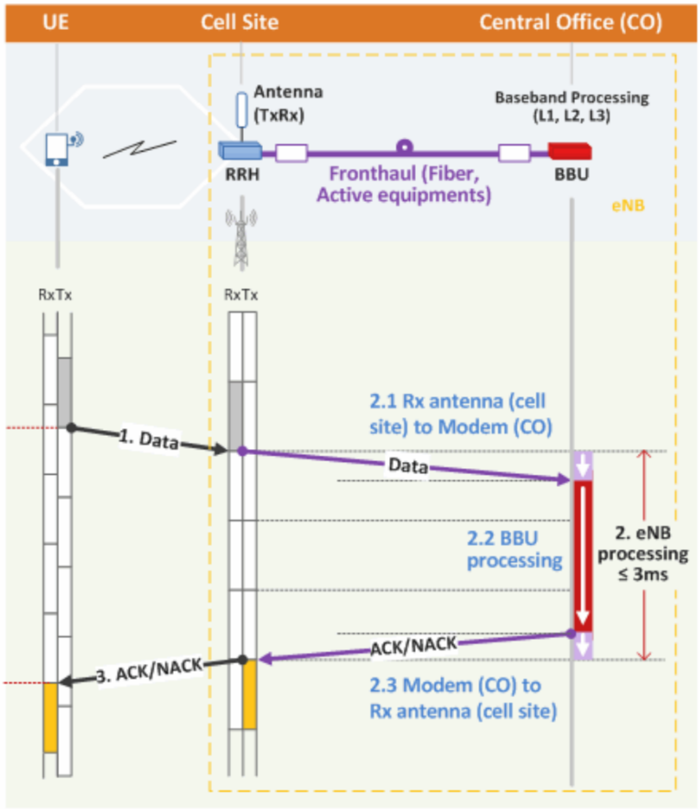


Figure 2.8: Fronthaul Impact on Latency requirements (retrieved from [Har14]).

The HARQ protocol states that the UE should receive a ACK/NACK (acknowledge/not acknowledge signal) from the eNB/BBU at least in the fourth subframe after the RRH receive the uplink data as illustrated in Figure 2.8. Since in LTE each subframe is 1 ms long this means that the fronthaul round trip transmission delay and baseband processing time in the BBU must be done in less than 3 ms.

A possible solution to reduce the required data rates and relax timing requirements in fronthaul would be to maintain some functions in the RRH side rather fully centralize all processing functions. This brings, however, some trade-offs since the higher level of processing maintained in the RRH translates into lower gains using CoMP technologies, more complex RRHs and consequently greater difficulty in maintaining and evolving the operator's network. More details are presented in the following chapter dedicated to the fronthaul. In the next chapter existing fronthaul solutions are explored and presented alternative solutions.

CHAPTER 3

Fronthaul

Dives into more detail in the fronthaul intricacies. Most commonly used protocols are briefly explained and alternatives presented.

3.1 Introduction

With the appearance of C-RAN architectures and the splitting of traditional base stations into BBUs and RRHs, a new digital link emerges to transport the data between those elements - the fronthaul. Although this link already exists in contemporary radio access networks it is not recognized as fronthaul. In its first instance this link was used at conventional RANs as an internal base station copper/coaxial link to connect the antennas on top of masts and buildings to the RF equipment located at the base. It was later replaced by an optical fiber connection transmitting digitized radio IQ samples in order to move the radio equipment next to the antennas and keep the base band processing at the base. The optical link provided non-existent electromagnetic interference and much lower transmission losses compared to the usual 3dB of attenuation (half the power) in the received power of electrical transmission.

As a result of splitting traditional base stations in BBUs and RRHs the demand is now to stretch the fronthaul link over several km's. Data is usually transferred under a digital radio over fiber standard or industry standard like CPRI or OBSAI. As most widely adopted fronthaul interface protocol, the following section describes the main features of the CPRI specification and its limitations that are driving the industry in search of alternative solutions.

3.2 CPRI

CPRI is an industrial agreement defined by Ericsson AB, Huawei Technologies Co. Ltd, NEC Corporation, Alcatel Lucent and Nokia Networks (as of Specification V7.0, 2015-10-09) [AB+15]. Aimed to define a publicly available specification for transport, connectivity and control to enable flexible deployment and interchangeable equipment from different vendors.

The specification focus on hardware dependent layers 1 and 2 and delineate the integration between Radio Equipment Controller (REC) and Radio Equipment (RE) by the use of three distinct information flows [AB+15] (Figure 3.1):

- User data;
- Control and management;
- Synchronization.

User plane information is transferred in the form of digitized IQ radio samples and time multiplexed with control and management data. In a fully centralized C-RAN architecture, the REC would be equivalent to the BBU and the RE to the RRH.

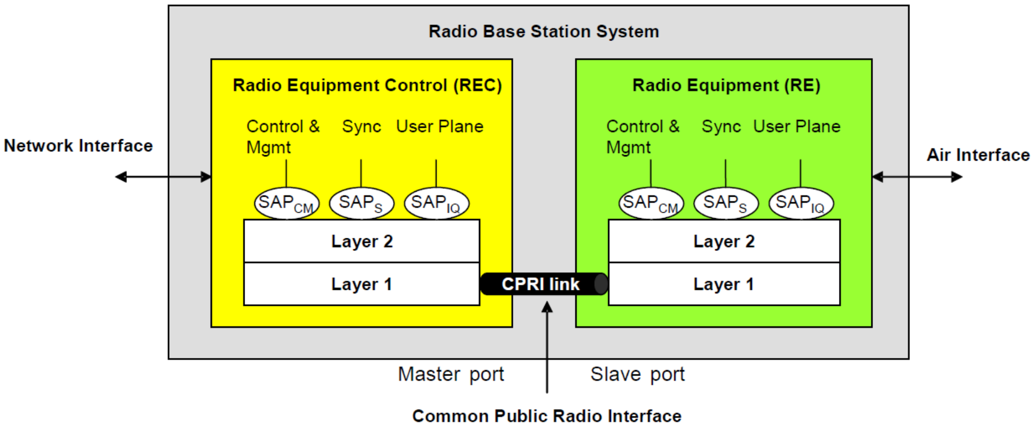


Figure 3.1: Basic System Architecture and CPRI Interface Definition (retrieved from [AB+15]).

Layer 1 establish the electrical/optical transmission and Time Division Multiplexing (TDM) mechanisms characteristics while Layer 2 specifies IQ data, control and management data (Ethernet/High level Data Link Control (HDLC)) and vendor specific data (Figure 3.2). To be noted that CPRI does not fully define the transmission specifications, whether electric or optical, as long as CPRI requirements are fulfilled. Typically, an optical fiber connection using SFP connectors are recommended by allowing higher flexibility in transceiver selection. SFP connectors allow for a wide range of transceivers, varying in maximum capable line bit-rate (up to tens of Gbps), ranges from a few meters to 100 km, several wavelengths and perhaps most importantly, cost.

CPRI operation is largely defined in units of the “chip rate,” defined in the 3rd Generation Partnership Project (3GPP) specifications as $T_C = 1/3.84$ MHz or 260.42 ns.

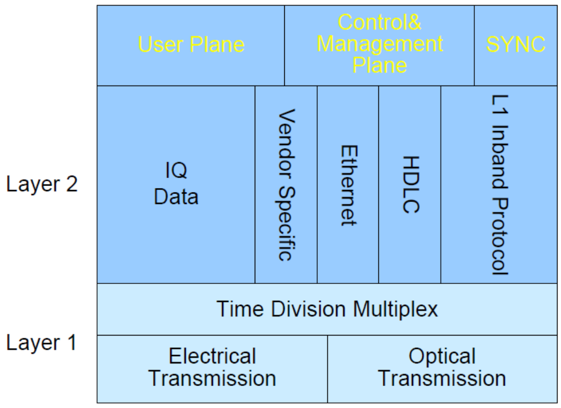


Figure 3.2: CPRI protocol overview (retrieved from [AB+15]).

3.2.1 Line Rate

CPRI has well defined line rates, chosen in order to facilitate the recovery of the basic UMTS chip rate of 3.84 Mbit/s. Table 3.1 resumes the line bit rate options and encoding scheme used. Note that Line Bit Rate option 7A carries the same information as option 7, only with a different line coding [AB+15].

Table 3.1: CPRI Line Bit Rate options and respective Word Length and Line Encoding Scheme

Line Bit Rate Option #	Line Bit Rate [Mbit/s]	Word Length T [bit]	Line Encoding
1	614.4	8	8B/10B
2	1228.8	16	
3	2457.6	32	
4	3072.0	40	
5	4915.2	64	
6	6144.0	80	
7	9830.4	128	
7A	8110.08		64B/66B
8	10137.6	160	
9	12165.12	192	
10	24330.24	384	

3.2.2 CPRI Frame Hierarchy

CPRI most elementary block is the basic frame represented in Figure 3.3. Each basic frame is formed by 16 words of length T, with T dependent of the line rate used. The first word of each basic frame represent the control word. Table 3.1 shows how the word length varies with the line rate. The control word share the same length except for line rate option 8, 9 and 10 in which the control word length is locked to 128 bits.

A basic frame length is defined as 1 T_C, so all IQ data and frame synchronization depend directly on this time constant. Basic frames are then arranged in groups of 256 to form an

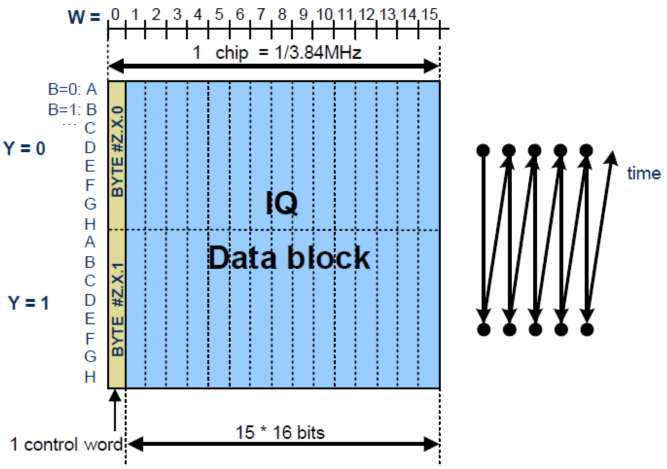


Figure 3.3: CPRI Basic Frame Structure for a 1228.8 Mbit/s line rate (retrieved from [AB+15]).

hyperframe. As a result each hyperframe is transmitted in 66.67 us. Lastly there is the CPRI 10ms frame that consists of 150 hyperframes (Figure 3.4). Synchronization information is incorporated in the control word of the first basic frame of each hyperframe.

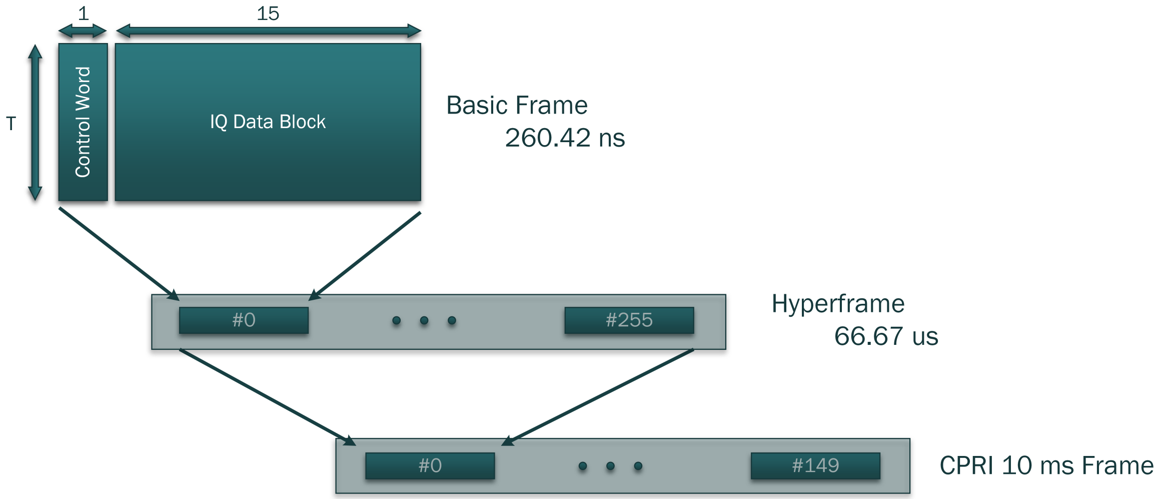


Figure 3.4: CPRI Frame Hierarchy.

3.2.3 Topologies

Although the most basic configuration (Figure 3.1) is composed of one REC and one RE connected by a single CPRI link, it can be extended in several topologies to increase capacity and support different RATs, especially considering that CPRI is a point-to-point link standard and requires that each CPRI link transfer only the IQ data flow of a single antenna/carrier. As such, multiple CPRI links can be used to increase capacity in case of multiple antennas/RATs. Possible configurations include chain, tree and ring topologies or a combination of the formers (Figure 3.5).

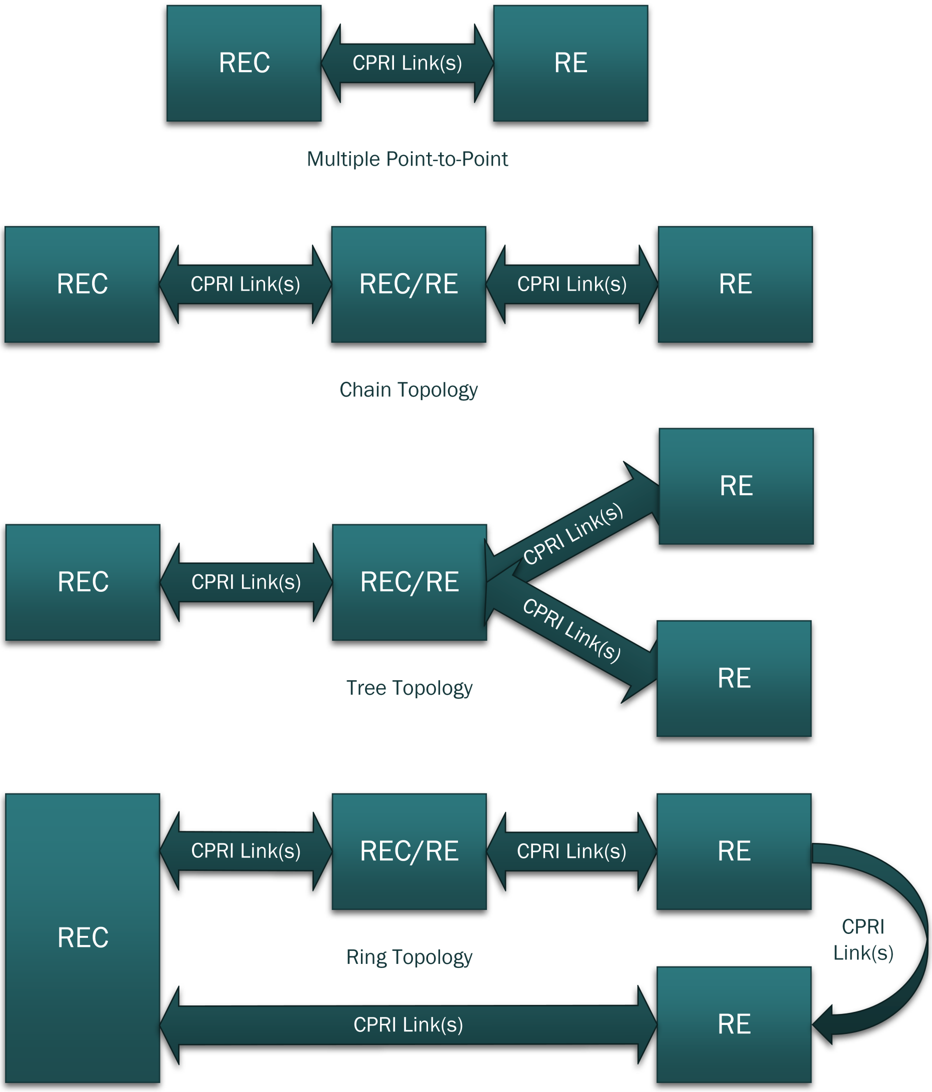


Figure 3.5: Possible CPRI topologies (adapted from [AB+15]).

3.2.4 Limitations

As a point-to-point link, CPRI framing is carried out at regular intervals with frame sizes matched to specific slices of wireless system frames in order to facilitate precise synchronization [Gom+15]. This means that the CPRI bandwidth increases proportional to the number of antennas. As such, C-RAN architectures with respective denser RRH network together with MIMO implementations and multiple RATs would lead to an huge amount of dedicated fibers and bandwidth, making large C-RAN deployment costly and inflexible. Furthermore, since the CPRI use a circuit oriented connection with constant line bit rate, no statistical multiplexing advantages can be obtained. Another consequence is the low transmission efficiency, since the line rate is antenna dependent, in other words, independent of active user data traffic. Additionally, CPRI has a one-to-one correspondence with the BBU that needs to be configured offline. This relationship can obstruct the underlying C-RAN paradigm of centralized processing and multiple remote radio heads. Lastly, CPRI compliant hardware from different manufacturers does not actually guarantee full interoperability, as it still contains vendor specific elements [Pfe15].

3.3 Evolution and Alternatives

In order to drive the industry to a more open and flexible C-RAN deployment, alternatives to current fronthaul standards are necessary. Ideally, the new fronthaul interface would need to be supported mainly by the following key features:

- Data rates decoupled from the number of antennas;
- Bandwidth adaptation to dynamic load;
- Support for multi-point links;
- Packet-based connections.

The main feature that would naturally bring out other advantages is a packet-based Ethernet solution. Packet switching optimizes the channel capacity by sharing the communication medium among other systems, increasing the overall network efficiency. By transmitting packetized fronthaul data, greater flexibility and scalability could be achieved as opposed to time division multiplexing.

Ethernet is now a mature technology as a result of years of development and its adoption within access networks could lead to a convergence with existing network systems. Ethernet interfaces are already completely mainstream in fixed network systems and standard IT servers, allowing easier integration with existing industry standard infrastructure and equipment. Furthermore, packet-based networks provide additionally protection and flexible routing capability.

On the other hand, a load dependent fronthaul interface would be capable of follow the dynamics of mobile subscribers and nullify the tidal wave effect mentioned in **Chapter 2, Mobile Networks**, greatly increasing the system efficiency.

From here, two natural approaches seem feasible: use Ethernet as the underlying transport of CPRI frames or carrying the radio samples directly over Ethernet frames. The first alternative would allow backward compatibility with legacy equipment with the implementation of simple CPRI mappers and de-mappers where needed. The disadvantage would be the additional framing overhead introduced as well as the remaining of one fundamental CPRI flaw - data rates dependent on the number of antennas rather than active user data traffic. Alternatively, the radio samples could be directly mapped in the Ethernet frames.

Although a packet-based fronthaul could answer many of the current CPRI limitations, it also comes with associated challenges. Wireless payloads are, by nature, very sensitive to synchronization, jitter and delay [Ins15]. An Ethernet based fronthaul would certainly lead to problems in these areas. Ongoing studies suggest that upcoming synchronization technologies such as Synchronous Ethernet (SyncE) or 1588v2 Precision Time Protocol could be part of the solution. These solutions are currently being developed and perfected by ITU-T and IEEE, respectively, in order to achieve synchronization mechanisms over received data or time-stamped packets instead of a local oscillator. Furthermore, in order to differentiate wireless payloads of other packet types, it may be necessary to introduce new encapsulation or protocol identifiers [Ins15].

3.3.1 eCPRI

Very recently, Ericsson AB, Huawei Technologies Co. Ltd, NEC Corporation and Nokia introduced a new fronthaul interface denominated enhanced CPRI (eCPRI) that is a direct evolution of the CPRI interface specification [AB+18a]. eCPRI emerges as a new packet based fronthaul interface in response to concerns from mobile network operators that the original CPRI standard wouldn't be able to fully support next generation use cases. The eCPRI interface specification encourages the use of Ethernet and IP technologies with the purpose of increasing overall transmission efficiency while supporting advanced networking and management. Furthermore, it possesses the ability to support various functional splitting points between the now denominated eCPRI Radio Equipment Control (eREC) and eCPRI Radio Equipment (eRE).

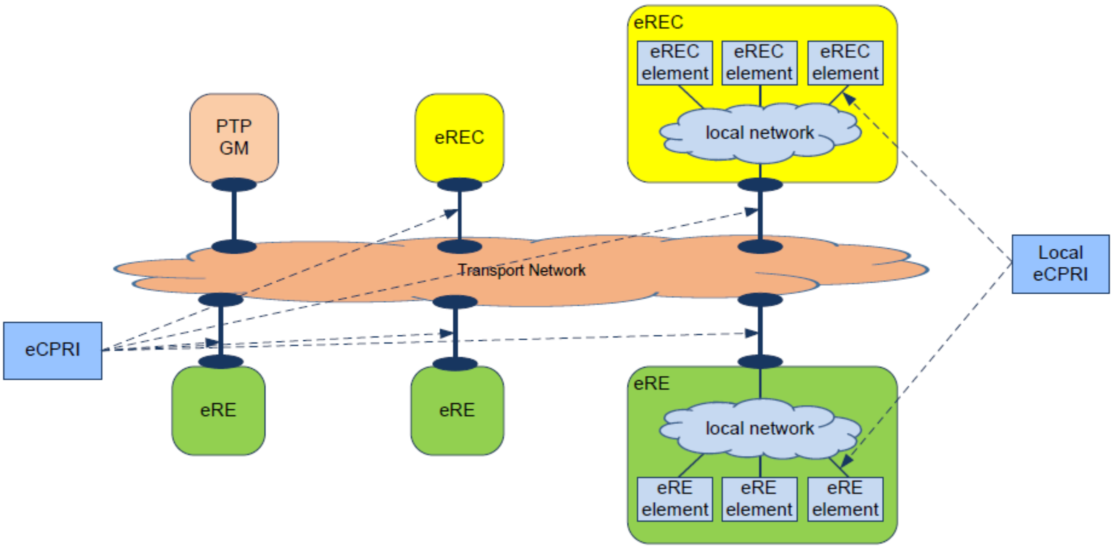


Figure 3.6: eCPRI system architecture (retrieved from [AB+18a]).

Figure 3.6 illustrates an eCPRI system architecture with several eREs and eRECs connected to a general transport network and local networks. The figure also represent a PTP GM, this is a grandmaster clock that serves as a reference clock to a Precision Time Protocol-based transport network. In the figure the GM is located in the network but it could be present in the eREC or eRE [AB+18a]. As in the previous CPRI specification, eCPRI also does not limit the use of network/data link layer protocols as long as it meets eCPRI requirements. An example of the eCPRI protocol stack over a IP/Ethernet network is shown in Figure 3.7.

eCPRI user plane messages follow the format illustrated in Figure 3.8. All messages contain a common 4 bytes long header followed immediately by the variable length payload. The

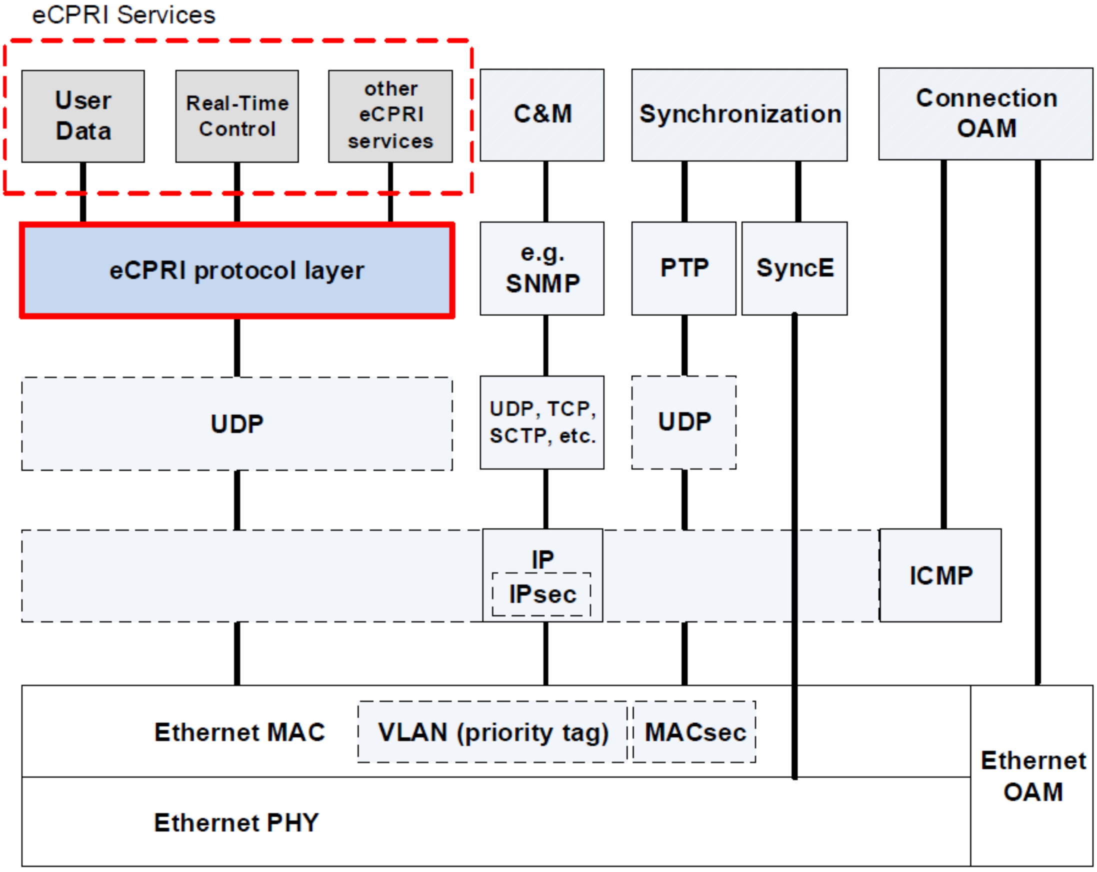


Figure 3.7: eCPRI protocol stack over IP/Ethernet (retrieved from [AB+18a]).

eCPRI Common Header accommodate the eCPRI protocol revision, message type, payload size, reserved bits and a single bit field indicating if the message is the last one inside the eCPRI protocol data unit. In the case of usage of Ethernet frames to transport the user data plane information, the Ethernet frame must use the eCPRI Ethertype 0xAEFE.

Contrary to the original CPRI, C&M and synchronization plane information is not defined in the eCPRI specification and therefore are not transmitted over the eCPRI specific protocol [AB+18a] but can be transported over any currently used protocol. In this case C&M plane could be transported over Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) for example and synchronization achieved over SyncE or Precision Time Protocol (PTP). Although, the eCPRI specification does not preclude other possible solutions. Synchronization however, primarily in the eRE side, must fulfill the requirements set by 3GPP over synchronization and timing accuracy. Table 3.2 shows some concrete requirements regarding one-way frame performance. Full network requisites are defined in [AB+18b].

The user data and synchronization planes are considered as the most time sensitive and as such should require a high priority while the control and management plane is considered non-time-critical and dealt with either lower priority or as background traffic.

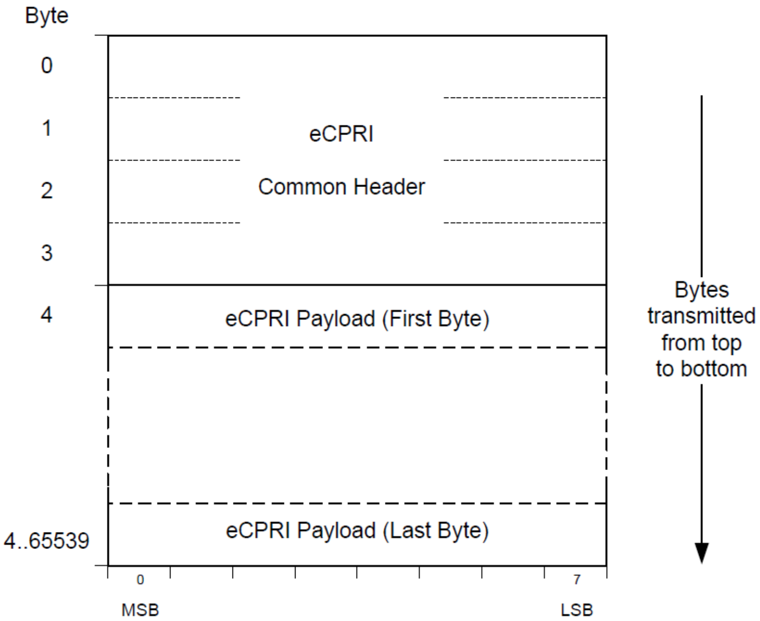

Byte
0
1
2
3
4
4..65539
eCPRI
Common Header
eCPRI Payload (First Byte)
eCPRI Payload (Last Byte)
0
7
MSB
LSB
Bytes transmitted from top to bottom

Figure 3.8: eCPRI message format (retrieved from [AB+18a]).

Table 3.2: eCPRI network requirements (adapted from [AB+18b]).

Priority	Use Case	Maximum One-way Frame Delay	Maximum One-way Frame Loss Ratio
High	User Plane (fast)	100 μs	10^{-7}
Medium	User Plane (slow) C&M Plane (fast)	1 ms	10^{-7}
Low	C&M Plane (slow)	100 ms	10^{-6}

3.4 Bandwidth Challenges

As mentioned briefly in **Chapter 2, Mobile Networks**, despite a fully centralized solution would clearly bring the most benefits regarding simpler RRHs and cooperative processing technologies at the BBU pool, it would also expose the fronthaul to a great burden concerning jitter, synchronization and, more importantly, bandwidth. Even with a new fronthaul interface, the required bandwidth and bit rates are still one of the main concerns regarding the fronthaul link. With 5G standards introducing higher frequency bands and maximum available bandwidths for the user, the required bit rates in the fronthaul link will grow exponentially and as such there is the need to come up with potential solutions capable of narrow down the required bandwidth in the fronthaul.

3.4.1 Function Splitting

First, keeping some low level processing in the RRH could decrease dramatically the required bandwidth and allow for more flexible timing constrains in exchange of increased complexity in the RRH when compared to a fully centralized solution. In particular, in order to eliminate the antenna dependent bit rate, it is suggested to keep antenna related functions in the RRH. These could include, Fast Fourier Transform (FFT)/Inverse Fast Fourier Transform (IFFT), resource mapping/de-mapping and channel estimation/equalization.

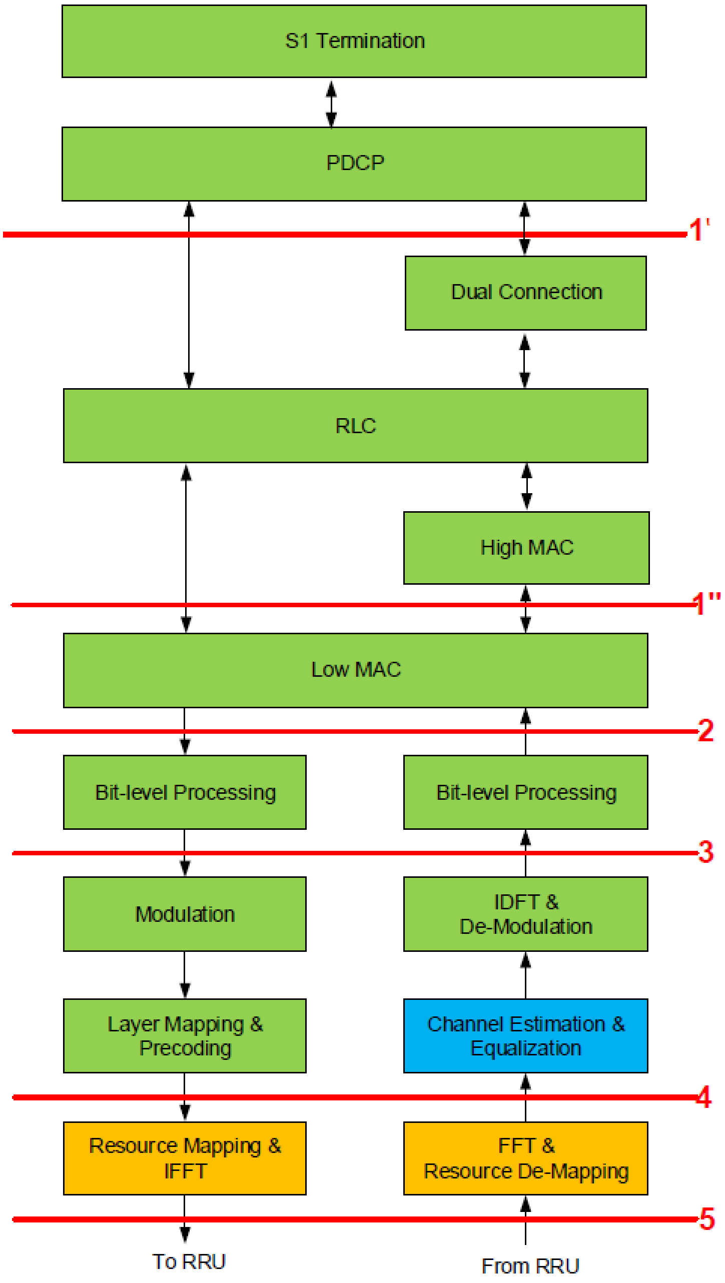


Figure 3.9: Functional splitting points (retrieved from [Ins15]).

Figure 3.9 represent potential function splitting points purposed by [Ins15]. Yellow blocks represent traffic-independent processing modules, while the remaining modules are dependent on user traffic. In this case each splitting point presents its advantages and disadvantages, influencing the necessary bandwidth, maximum allowed delay and supported cooperative technologies. In this scenario the splitting point number 5 is equivalent to a fully centralized solution.

Table 3.3: Fronthaul bandwidth requirement VS splitting point (adapted from [Ins15]).

	Peak Bandwidth				
	Split 1	**Split 2**	**Split 3**	**Split 4**	**Split 5**
Downlink	174 Mb/s	179.2 Mb/s	125.2 Mb/s	498 Mb/s	9830.4 Mb/s
Uplink	99 Mb/s	78.6 Mb/s	464.6 Mb/s	2689.2 Mb/s	9830.4 Mb/s

From Table 3.3 it is shown that the required bandwidth decreases considerably as we move away from a fully centralized solution (split 5 -> split 1). This brings, however, some trade-offs since the higher level of processing maintained in the RRH translates into lower gains using cooperative technologies, more complex RRHs and consequently greater difficulty in maintaining and evolving the mobile operator's network. It is also important to note that in Table 3.3 while the bandwidths between solutions 1-4 are peak values, in solution 5 the bandwidth is constant. These numbers were calculated assuming a single 20 MHz LTE carrier, 2 antenna ports with 8 antennas, 64 QAM for downlink and 16 QAM for uplink and maximum number of users is 100.

Table 3.4 summarize the fundamental pros and cons of the various functional splitting points. The most advantageous split points appear to be solutions 3 and 4 that allow significant reductions in bandwidth while keeping RRHs relatively simple and high gains in cooperative technologies. For example, by splitting functions by mark 4 (Figure 3.9) the required bandwidth can be decreased by a factor of 20 in the downlink and a factor of 3 in the uplink.

3.4.2 Compression

On the other hand, compression techniques can also significantly reduce the total required bandwidth without sacrificing noticeable performance. According to [Anj+15], by using non-linear quantization techniques based on FPGA Look Up Tables (LUTs), more than 40% compression ratio can be achieved with a negligible Error Vector Magnitude (EVM) degradation of 0.35%. Latency overhead is minimal as the compression/decompression relays entirely on simple combinatorial logic.

Table 3.4: Summary of different split point advantages/disadvantages (adapted from [Ins15]).

	Split 1	**Split 2**	**Split 3**	**Split 4**	**Split 5**
RRH Complexity	High	High	High	Low	Very Low
Pooling Gain and # of Supported Cooperative Technologies	Very Low	Low	Low	High	High
Complexity involved in Upgradability and Maintenance	High	High	High	Low	Low

3.4.3 5G Requirements

With the imminent introduction of 5G, the requirements for the fronthaul segment will be even more extreme. Ongoing standardization by 3GPP points to the use of much higher frequency carriers in sub-6 GHz and mmWave bands along with ultra wide carrier bandwidths in the order of several hundred MHz. These factors coupled with massive MIMO antenna configurations will impact the required fronthaul bandwidth linearly, further emphasizing the idea that a fully centralized solution transmitting IQ radio samples is simply not feasible in the long run to support 5G use cases. Studies conducted in [Bar+17] show that, depending on the split points, data rates can go from 20 Gbps to almost 200 Gbps, as observed in the graph to the left of Figure 3.10.

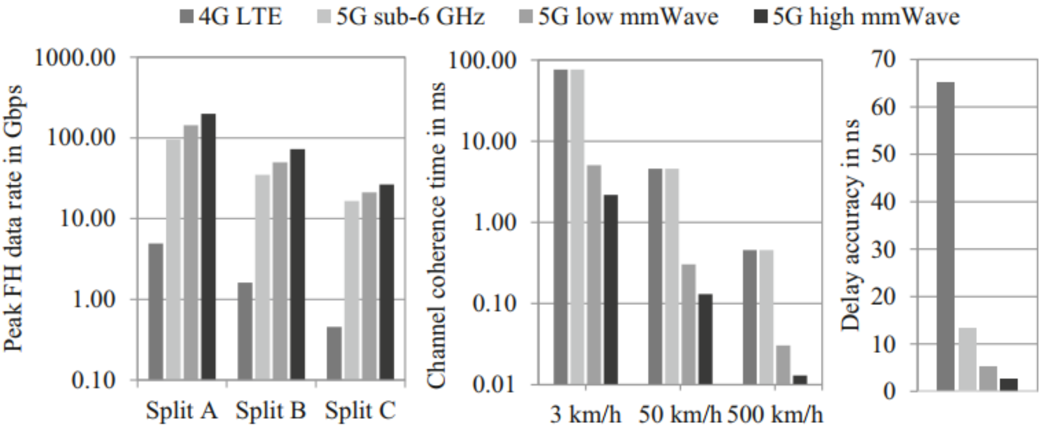


Figure 3.10: 5G fronthaul extrapolated requirements (retrieved from [Bar+17]).

This study was based on data from existing LTE networks and then extrapolated to 5G scenarios that consider radio access technologies and frequency bands that align as much as possible with current ongoing standardization. Table 3.5 shows some of the parameters used to determine forecast data in Figure 3.10.

Table 3.5: Parametrization of a 5G radio access network using different frequency bands (adapted from [Bar+17]).

Parameter	**Unit**	**4G**	**5G**		
		LTE	**Sub-6**	**Low mmWave**	**High mmWave**
Carrier Frequency	**GHz**	2	2	30	70
Channel Size	**MHz**	20	100	250	500
Sampling Rate	**MHz**	30.72	150	375	750
# Antennas	**-**	4	96	128	256
Quantizer Resolution Time Domain	**bit**	15	15	12	10
Modulation Order	**-**	64	1024	256	64
Frame Duration	**ms**	1			
FFT Size	**-**	2048			
# Active Subcarriers	**-**	1200	1300		

In this case, split A corresponds to a fully centralized solution, split B keeps FFT/IFFT and Resource Mapping/De-mapping in the RRU and finally split C completely divides PHY and MAC. Split A, B and C correspond respectively to split 5, split 4 and split 2 of Figure 3.9. In addition, the rightmost graph also denotes a significant reduction in the allowed delay accuracy.

The next chapter presents a C-RAN test-bed using a 20 km CPRI-based fronthaul currently deployed in Instituto de Telecomunicações - Aveiro. This test-bed will later act as the foundation for the 10 Gbit Ethernet based fronthaul.

CHAPTER 4

CPRI based C-RAN Test-bed

Introduces the CPRI-based C-RAN laboratory test-bed: physical infrastructure, major logical blocks, and functionality. This is the underlying infrastructure which will serve as a base to the Ethernet based fronthaul implementation.

4.1 Introduction

The Flexicell test-bed is a fully operational C-RAN laboratory test-bed developed in a previous project (Flexicell) at Instituto de Telecomunicações - Aveiro. In its current state, the test-bed is set-up to implement an LTE network accordingly to a fully centralized C-RAN topology, with a single RRH connected to a single BBU in the processing pool. It features a CPRI-based optical fronthaul interface with a 20km extension and Passive Optical Network (PON) compliant infrastructure. Open Air Interface (OAI), a industrial/academic community driven open source software alliance, emulates the full software implementation of the 3GPP LTE protocol stack for both core and access networks [Ope17]. The RAN can cover several LTE bands in a small radius and support commercial user equipment like smartphones and USB dongles.

4.2 System Architecture

The radio access network consists primarily by the EPC, BBU, fronthaul and RRH as illustrated in Figure 4.1).

The RRH, composed by the RF front-end and a KC705 evaluation board featuring a Kintex-7 FPGA, connects to the BBU via a 20 km PON fronthaul. Being a fully centralized solution based on the CPRI specification, user data is exchanged in the form of digitized IQ radio samples between base band unit and remote radio head. The BBU itself consists of a general purpose processor based machine running the LTE soft-modem and a VC707 evaluation board featuring a Virtex-7 FPGA interfacing with the optical fronthaul. The BBU

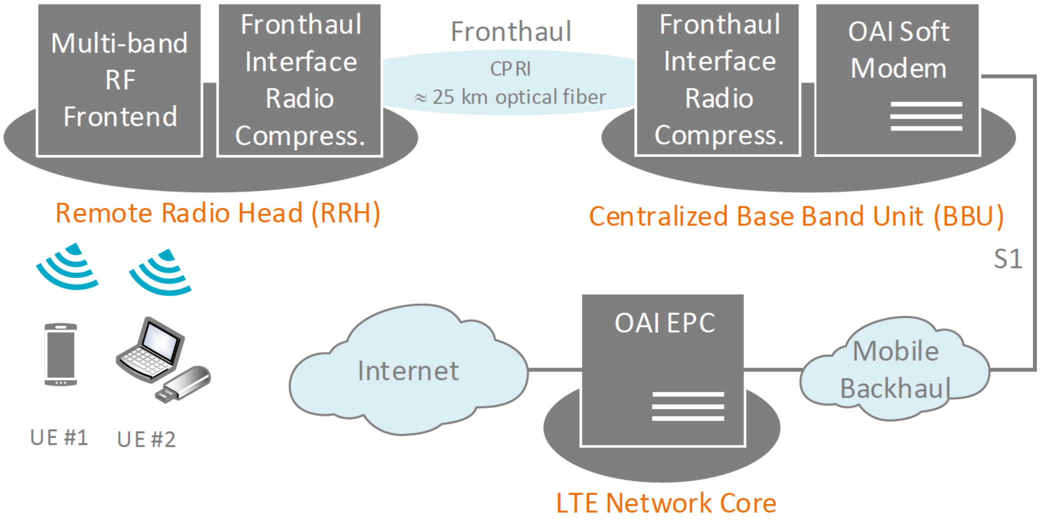


Figure 4.1: CPRI based C-RAN system architecture (retrieved from [San+16b]).

is connected to the core network (EPC) by the S1 link carrying both user and control plane data using a dedicated Gigabit Ethernet over copper connection.

4.3 Open Air Interface

The OAI platform is an open source software and hardware development for the core and access network of 3GPP cellular networks standards [Ope17]. The intelligence of the network: EPC and LTE soft-modem, use OAI-based baseband processing and core network virtualization for a totally open and flexible test-bed. The system enables full interoperability with Commercial Off-The-Shelf (COTS) devices. Figure 4.2 shows the implemented LTE protocol stack according to 3GPP.

Open Air Interface makes use of general purpose processors for processing power and SDR front-end architectures in order to easily reconfigure the RRH on the fly. The open

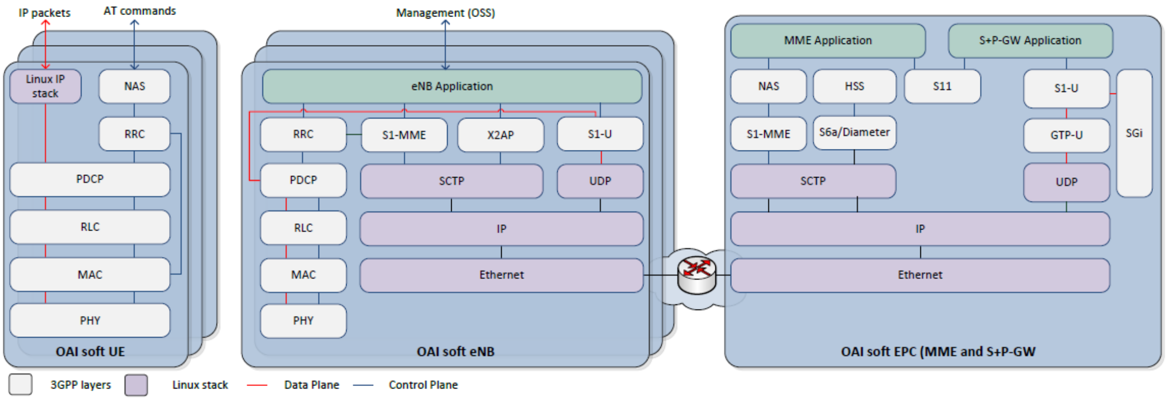


Figure 4.2: OpenAirInterface LTE protocol stack (retrieved from [Ope17]).

source software is based on standard C and developed for real-time Linux kernels running on top of Intel x86 or ARM general purpose processors under the OAI License Model. Low latency kernels are required, along with the CPU running with a locked frequency and power management features/frequency scaling turned off to avoid real-time issues.

Besides the full interoperability between OAI platforms, different configurations involving commercial components (EPC, eNB and/or UE) are also supported.

OAI official support regarding SDR platforms is limited to 3rd party units such as the EXMIMO2, X-series/B-Series USRPs, BladeRF and LimeSDR that connect directly to the OAI running computer through Peripheral Component Interconnect express (PCIe) or Universal Serial Bus (USB).

In order to achieve a C-RAN deployment, instead of 3rd party SDR solutions connected directly to the BBU, a VC707 evaluation board was used with custom drivers specially developed to ensure proper connectivity and compatibility to the OAI eNB via PCIe. The VC707 then provides all the required connectivity for the fronthaul while another FPGA-based evaluation board (KC705) implements the RRH.

4.4 Physical Infrastructure

The infrastructure is mainly supported by FPGA based evaluation boards by Xilinx (VC707 and KC705) and a couple of machines relying on general purpose processors. These evaluation boards provide SFP connectivity to back the optical fronthaul, Ethernet, FPGA Mezzanine Card (FMC) and SubMiniature version A (SMA) connectors for enhanced connectivity and expansion capabilities. Additionally, the FPGA based system enables a flexible co-development environment for both hardware and software [San+16a].

4.4.1 BBU

Figure 4.3 illustrates the BBU block diagram. The BBU is composed by a physical machine and a Virtex-7 FPGA based VC707 evaluation board. The physical machine is running an Intel i5-4440 CPU locked at 3.1 GHz with all power management and frequency scaling features disabled. The VC707 development kit implements the CPRI link, interfaces with the optical fronthaul link through a SFP Dense Wavelength Division Multiplexing (DWDM) transceiver, performs IQ data compression and decompression using non-linear quantization techniques based on Look Up Table (LUT) of the IQ samples and transfers them to the physical machine through a PCIe x8 Gen2 connection directly to the physical machine motherboard. The computer then implements the OAI LTE soft-modem under a Ubuntu distribution with a low-latency kernel.

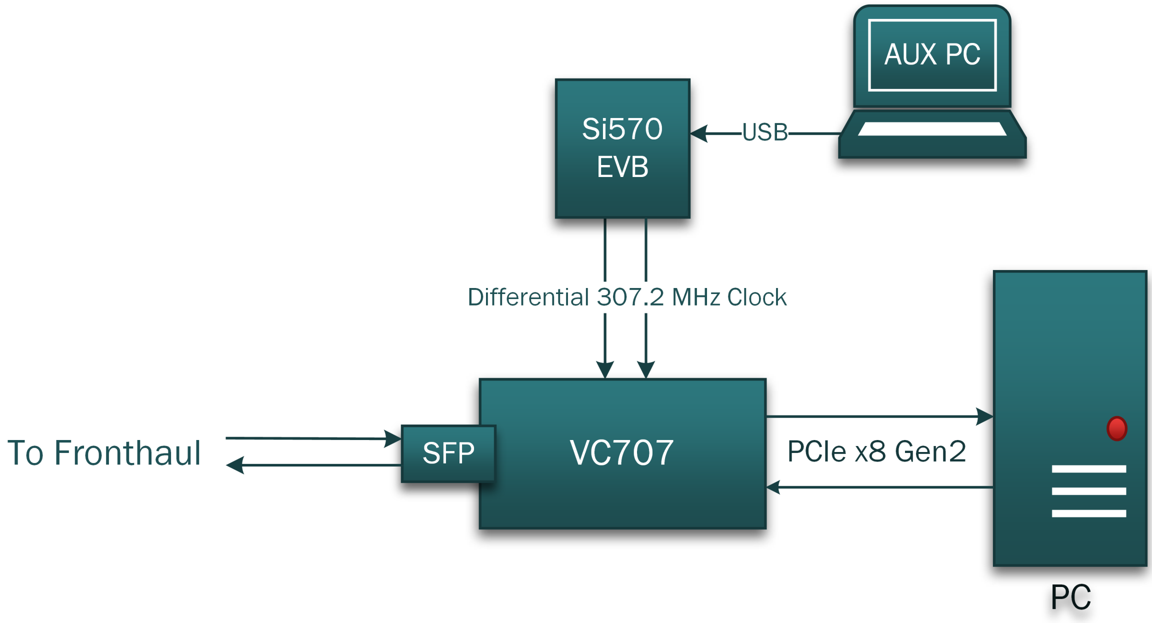


Figure 4.3: BBU block diagram.

The required CPRI link differential reference clock of 307.2 MHz is generated externally on a Si570 evaluation board (Figure 4.4). The Si570 is a programmable crystal oscillator capable of output frequencies ranging from 10 to 945 MHz and selected frequencies up to 1.4 GHz [Lab14b]. The kit is powered through USB and allow output frequency programming by the proprietary Programmable Oscillator Software [Lab14a]. The differential clock is delivered to the VC707 kit by SMA connectors.

Figure 4.4: Si570 Evaluation Board.

4.4.2 Fronthaul Interface

Connecting the BBU and RRH is a 20 km fully passive optical fronthaul making use of the CPRI standard to exchange user data, control and management messages and synchronization. Figure 4.5 illustrates the fronthaul block diagram and optical transmission details.

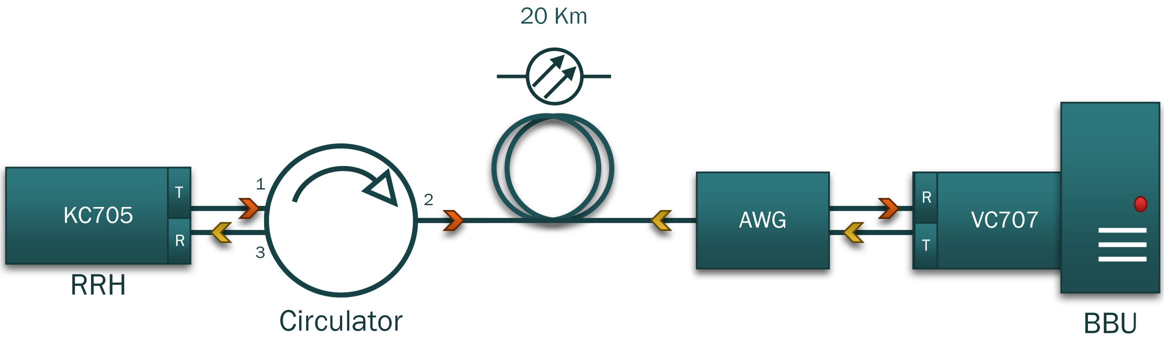


Figure 4.5: Fronthaul block diagram.

The optical infrastructure is comprised as follow: at the "edge" of the base band unit, connected to the VC707, are the SFP electro/optical transceiver like the one presented in Figure 4.6.

Figure 4.6: Small Form-factor Pluggable (SFP) transceiver (retrieved from [Fin09b]).

An SFP is a bidirectional device with the transmitter and receiver in the same physical package. It converts the electrical signals from the gigabit transceivers in the board to the optical domain and vice-versa. Optical transmission is achieved under a simplex or duplex LC connector and single mode optical fibers (Figure 4.7). Currently used SFPs are FWLF-1631-xx DWDM transceivers from Finisar capable of reaching peak bit-rate of 2.7 Gbps (support up to CPRI line bit-rate option 3) and a maximum range of 120 km over single mode optical fibers with wavelengths of 1542.14 nm (RRH) and 1530.33 (BBU) [Fin09b].

It then passes through a Arrayed Waveguide Grating (AWG) where multiple wavelengths are multiplexed to achieve bi-directional transmission over a single fiber. In this particular case just two wavelengths are combined: the receiver and transmitter wavelength. Exiting the AWG the optical signal travel through two 10 km optical fiber coils and enters a 3-port optical circulator. In a circulator each wavelength entering one port is transmitted to the next port. In this particular case, one port is used as input, one as output and the third

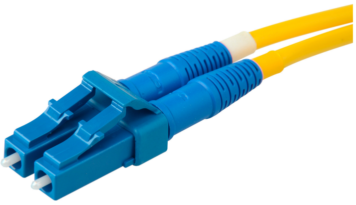


Figure 4.7: LC duplex connector with yellow marked single-mode optical fibers.

used as a bi-directional port to transmit the previous two signals in opposite directions in a single optical fiber. The split optical signals finally connect to the RRH board via an SFP transceiver as described previously. In order words, the 1530.33 nm wavelength transmitted by the VC707 (symbolized by yellow arrows in Figure 4.5) is multiplexed in the AWG, travels through two 10 km optical fiber coils, enters port 2 of the circulator and exits the circulator through port 3, reaching the KC705 receiver. In the opposite direction, the 1542.14 nm wavelength transmitted by the KC705 (symbolized by orange arrows in Figure 4.5) enters port 1 of the circulator and exits in port 2, travels through the optical fiber coils, is de-multiplexed in the AWG and finally enters the VC707 receiver.

With this kind of setup the test-bed have the possibility to accommodate other Wavelength Division Multiplexing (WDM) compliant systems.

4.4.3 RRH

A complete block diagram of the RRH is shown in Figure 4.8.

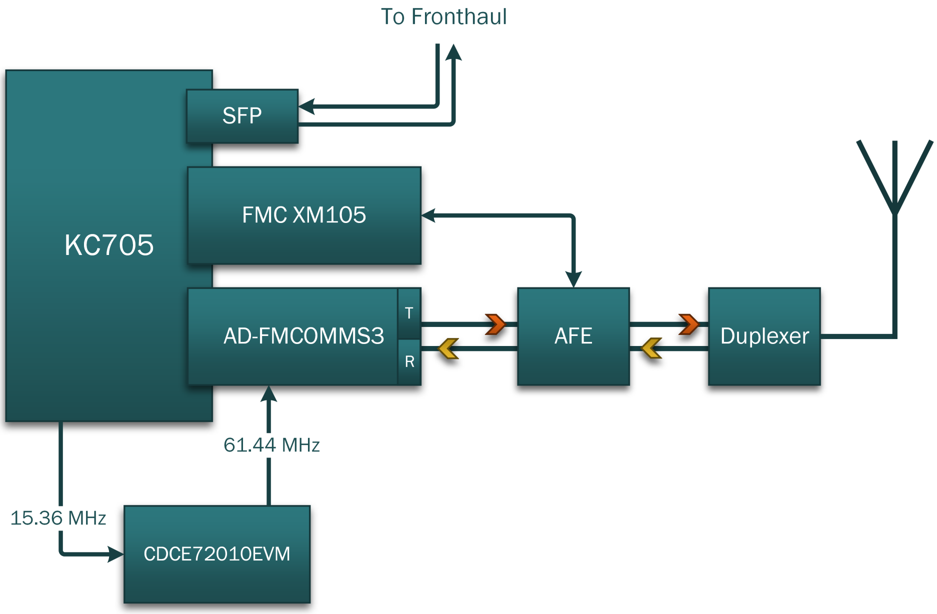


Figure 4.8: RRH diagram block.

The remote radio head can be divided in the digital and analog RF front-end. Both are controlled by a KC705 development kit by Xilinx (featuring a Kintex-7 FPGA) through its FMC connectors. The multi-band Analog Front-End (AFE) provides communication over the air interface to transmit and receive modulated radio signals. It consists mainly of power amplifiers, filtering stages, low-noise amplifiers and switches capable of covering LTE bands 1, 3, 7, 8 and 20 (Figure 4.9). The AFE is controlled through a FMC XM105 daughter board that provides multiple input/output pins, headers and a pair of SMA connectors.

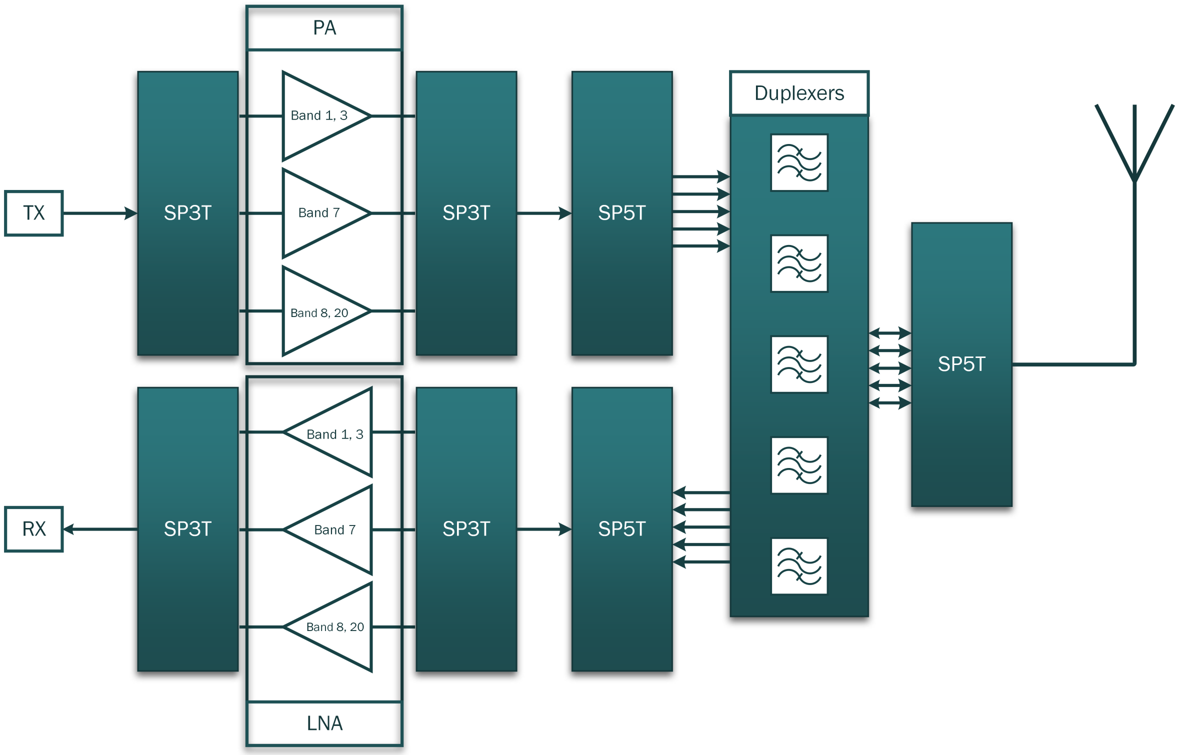


Figure 4.9: Multi-band analog RF front-end.

The digital front-end SDR solution is based on a AD-FMCOMMS3-EBZ evaluation board by Analog Devices [Ana17a]. The AD-FMComms3-EBZ is an High Pin Count (HPC) FMC daughter board for the AD9361 - an integrated RF transceiver intended for use in SDR applications. The AD9361 is tunable from 70MHz to 6GHz with a channel bandwidth up to 56MHz and two TX/RX channels [Ana17b].

The KC705 also provides connectivity with the optical fronthaul through a SFP DWDM transceiver.

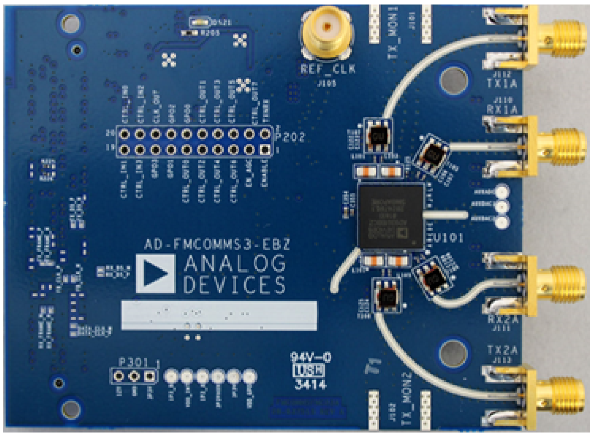


Figure 4.10: AD-FMCOMMS3-EBZ FMC daughter card. Top side, with visible AD9361 integrated transceiver (retrieved from [Ana17a]).

4.4.3.1 Clock Recovery Circuit

The references clock required for the radio front-end is recovered from the fronthaul CPRI link and ran through a Silicon Labs Si5324 jitter attenuator on the back of the KC705 that feeds the "cleaned" clock signal back to the transceiver and clock recovery circuit. Since the AD9361 needs an extremely precise clock signal, the clock recovered previously is fed to an external low jitter clock synchronizer, the CDCE72010, present in a CDCE72010EVM evaluation module from Texas Instruments to attenuate the jitter even further, before being redirected to the AD-FMCOMMS3-EBZ internal Phase Locked Loop (PLL).

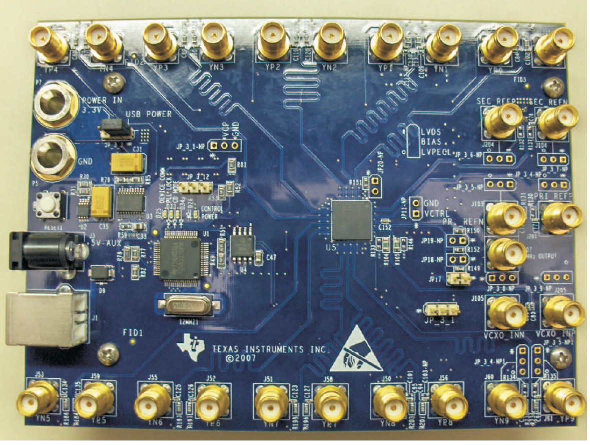


Figure 4.11: CDCE72010EVM evaluation board featuring a CDCE72010 low jitter clock synchronizer (retrieved from [Tex08]).

The CDCE72010EVM shown in Figure 4.11 features two input references and through its on-chip PLL can provide up to 10 differential or 20 single-ended low jitter outputs. The output clocks support several physical layer specifications such as Low Voltage Differential Signaling (LVDS), Low Voltage Positive Emitter-Coupled Logic (LVPECL) and Low Voltage Complementary Metal Oxide Semiconductor (LVCMOS). The low jitter clock synchronizer is configured through USB by the graphical user interface (GUI) software and although it can be powered by the USB cable for small applications it is recommended to use and external 3.3V power supply [Tex08]. Figure 4.12 illustrate the RRH clock recovery circuit.

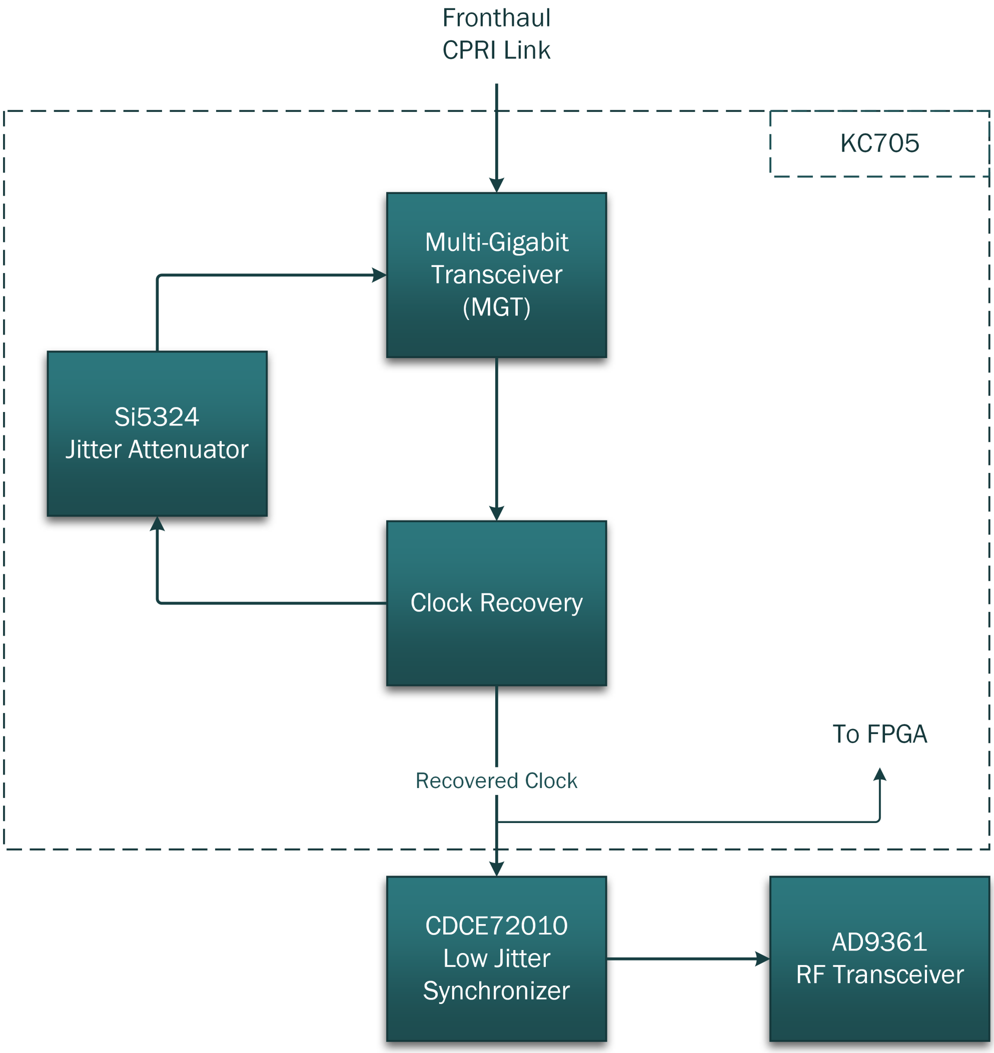


Figure 4.12: RRH clock recovery circuit.

4.4.4 EPC

Lastly there's the core network, in this case being a LTE network, the Evolved Packet Core (EPC). It is connected to the BBU physical machine motherboard by a standard Gbit Ethernet copper cable while the EPC itself is running on a virtual machine in the server room. The virtualized core network runs on top of a standard Ubuntu Linux distribution and it is provided by the OAI OpenCN software. The EPC main elements run concurrently as separate processes on the virtual machine: Mobility Management Entity (MME), Home Subscriber Server (HSS) and SPGW (comprising both Packet Data Network Gateway (P-GW) and Serving Gateway (S-GW)).

4.4.5 UE and SIM Cards

Figure 4.13: Different types of user equipment used in the RAN.

The setup supports commercial devices such as smartphones, USB dongles and 4G/LTE routers capable of providing mobile broadband to WiFi capable devices. In order to connect COTS devices to the test-bed, commercial SIM cards cannot be used as they are usually tied to the respective mobile network operator. UEs connected to the test-bed use special programmable SIM cards with the network information. The SIM card is then registered in the core network HSS mySQL database in order for the UEs to attach correctly.

4.5 Digital Designs

In this section is described in greater detail the internal logic designs implemented by the FPGAs in both KC705 and VC707 evaluation boards.

4.5.1 RRH

Starting by interfacing the RF front-end data interface is a pair of FIFOs (First-In-First-Out) each with independent read/write clocks in order to exchange data reliably from the front-end clock domain and the CPRI clock domain. Already in the CPRI clock domain side, digitized IQ radio samples go through a compression/decompression subsystem implementing non-linear quantization based on LUTs. Following compression/decompression, IQ samples are mapped/de-mapped by the E-UTRA module and delivered to the CPRI IP core provided by Xilinx, in charge of framing/de-framing the IQ radio samples and transmit/receive them to/from the optical link through the FPGA built-in Multi-Gigabit Transceivers (MGTs) and SFP transceivers.

An FPGA MGTs are responsible for serialized data transmission at bit-rates reaching multiple Gbit/s and usually also implementing clock data recovery, line encoding/decoding, alignment, equalization, serializer/deserializer, etc.

The CPRI core is also responsible for insertion/removal of control and management data flows to/from the Ethernet MAC core using the Gigabit Media-Independent Interface (GMII) interface. The Ethernet MAC core then passes the control messages to the embedded MicroBlaze soft-processor. Finally, the soft processor controls both analog and digital RF front-ends through an Serial Peripheral Interface (SPI) link.

Figure 4.14 illustrates the remote radio head FPGA block diagram.

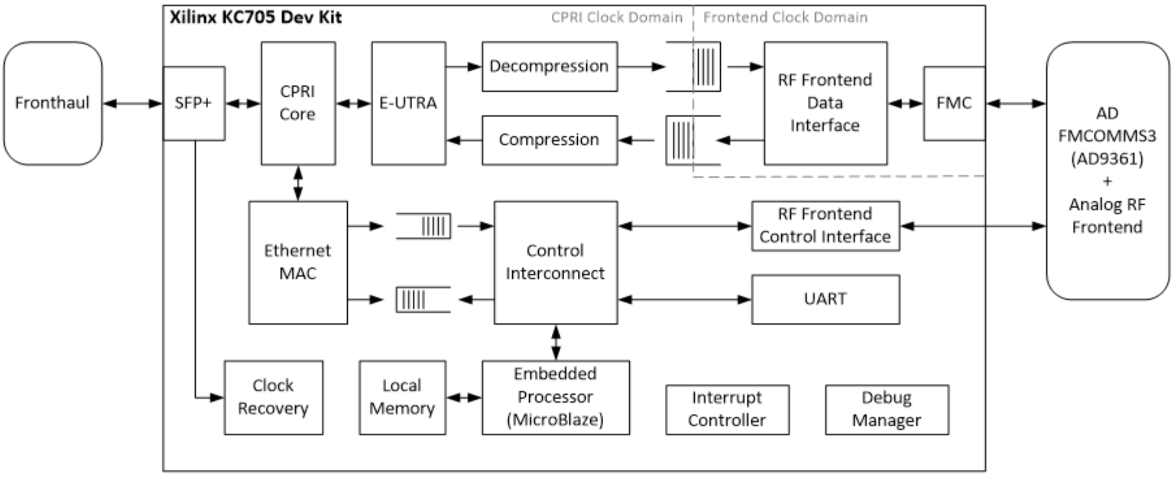


Figure 4.14: RRH logic block diagram (retrieved from [Oli16]).

4.5.2 BBU

On the BBU side, data path in the CPRI domain clock is very similar to the RRH side. Data travels through the CPRI core, E-UTRA module and compression/decompression subsystem. Data from the CPRI clock domain interfaces with the PCIe clock domain through block RAMs. Data stored in the block RAMs is transferred between the main computer running the LTE soft modem through a PCIe x8 Gen2 interface. Direct Memory Access (DMA) controllers manage the data transfer. Another embedded MicroBlaze soft-processor is in charge of board initialization and set-up as well as trading control and management flows with the CPRI core. The BBU block diagram is displayed at Figure 4.15.

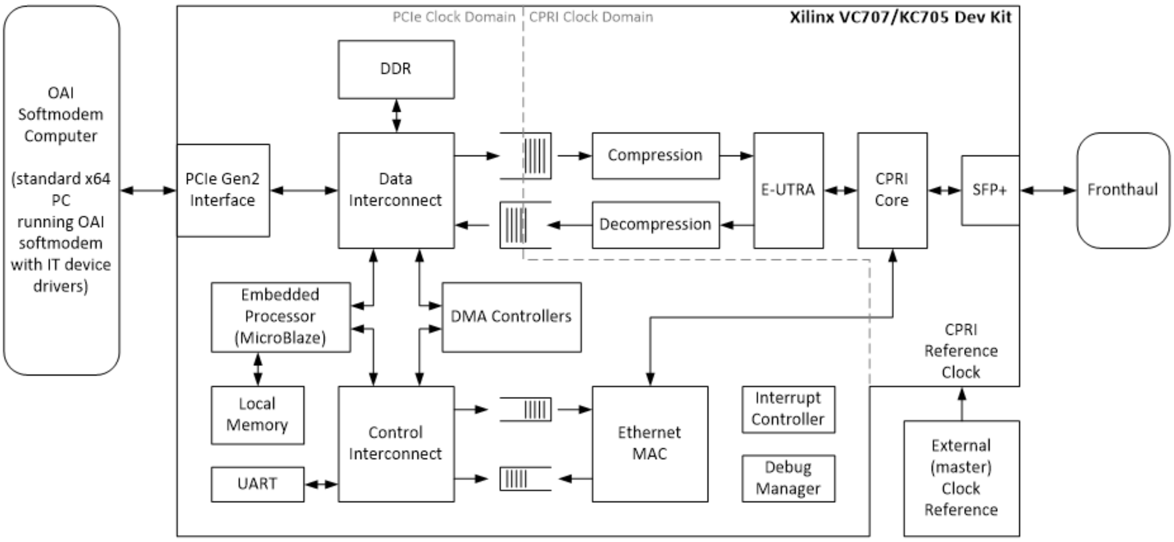


Figure 4.15: BBU logic block diagram (retrieved from [Oli16]).

4.6 MicroBlaze and Embedded Application

Running on both RRH KC705 and BBU VC707 boards is a MicroBlaze soft-processor. It's called a soft-processor since it is implemented using FPGA logic instead of dedicated hardware as in traditional processors. The MicroBlaze is provided by Xilinx as an IP core and is instantiated through the Vivado IP catalog.

On both boards, the MicroBlaze is used for control and management features by the means of a embedded application developed in Xilinx's SDK environment in standard C. The embedded application functions include:

- BBU VC707
  - Receive commands from the OAI LTE soft-modem running on the main computer through PCIe;
  - Initialize DMA controllers for data transfer between the main computer and the VC707 board through the PCIe interface;
  - Configure and initialize the CPRI IP core;
  - Configure and initialize the AXI Ethernet subsystem;
  - Send RRH configuration through the AXI Ethernet subsystem;
- RRH KC705
  - Configure and initialize the CPRI IP core;
  - Configure and initialize the AXI Ethernet subsystem;
  - Receive RRH configuration from the Ethernet subsystem;
  - Initialize UART;
  - Initialize and configure the AD9361 RF transceiver based in the RRH configuration received from the BBU;

This chapter marks the transition between the introductory documentation and theoretical concepts of earlier chapters to the physical implementation of the new fronthaul interface described in the following chapters where the CPRI specification is replaced by a 10 Gigabit Ethernet link.

CHAPTER 5

10 Gigabit Ethernet Link Implementation

This chapter starts by describing the development stages of this work followed by the 10 Gb/s Ethernet link implementation in FPGA.

5.1 Introduction

After initial considerations and taking into account the CPRI alternatives in **Chapter 3, Fronthaul** and the capabilities of the development boards available, it was proposed to implement the new fronthaul interface based on a 10 Gigabit Ethernet link over optical fiber. From here, two solutions were deemed possible:

- Encapsulate CPRI over Ethernet;
- Radio over Ethernet.

From the two options, it was decided to proceed with Radio over Ethernet, where IQ radio samples are directly mapped in Ethernet frames. This would ensure lower overhead latency and higher transmission efficiency while allowing for more control over the whole setup. Furthermore, currently used CPRI IP cores that require evaluation licenses (with limited continuous operation time) are no longer needed.

In order to achieve the final objective of this dissertation, that is, develop a fully functional 10 Gigabit Ethernet-based fronthaul and validate it on a working C-RAN test-bed, the following development stages where adopted:

1. Study and familiarization with the current C-RAN test-bed.
2. Development of a standalone 10 Gigabit Ethernet link between two FPGA-based development boards.
3. Integration with the existing C-RAN test-bed.
4. Validation, optimization and testing.

The first stage can be subdivided into two further substages. First, the study of the hardware infrastructure and digital designs of both RRH and BBU FPGAs platforms presented in **Chapter 4, CPRI based C-RAN Test-bed**. Secondly, a fresh installation of the Open Air Interface platform was performed since the former is deeply integrated with the C-RAN test-bed. This would allow for a familiarization with platform usage and how it interacts with the C-RAN, while at the same time would provide the opportunity to keep the old installation as backup.

From this first interaction with the C-RAN test-bed resulted **Appendix A, Flexicell Start-Up User Guide**, a detailed user guide on how to successfully start-up the original laboratory radio access network. Open Air Interface (OAI) installation is detailed in **Appendix B, OpenAirInterface Deployment**.

The remaining of this chapter focus on the second stage - development of the standalone 10 Gigabit Ethernet implementation.

5.2 Ethernet Controller Architecture

A typical Ethernet architecture is composed by several blocks arranged as seen in Figure 5.1. For this purpose let's group these blocks into three groups: physical layer, data link layer and upper layers.

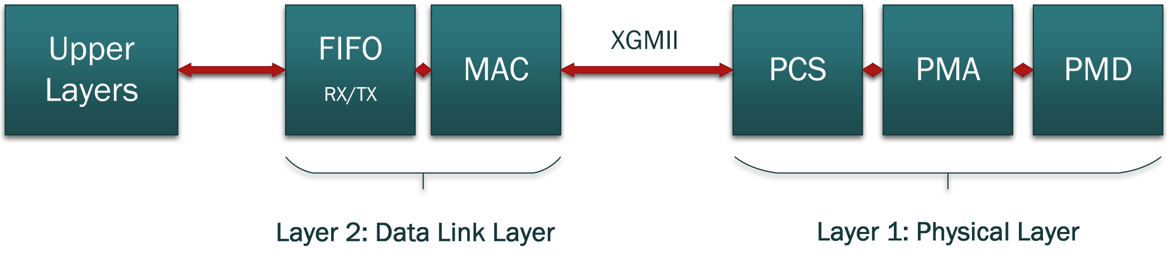


Figure 5.1: Ethernet System Typical Architecture.

The data link layer, represented by the Medium Access Control (MAC) and associated RX/TX FIFOs, is responsible for flow control, frame formatting/decoding and respective transmission/reception to the PHY through the media-independent interface (MII). Media-independent interface defines the interconnection between MAC and PHY and today's variants include GMII, Reduced Gigabit Media-Independent Interface (RGMII), Serial Gigabit Media-Independent Interface (SGMII) and in the case of 10 Gigabit Ethernet, 10 Gigabit Media Independent Interface (XGMII).

As the name suggest, the physical layer deals with the intricacies of transmitting and receiving raw bits over a physical transmission medium and can be subdivided in Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) and Physical Medium Dependent (PMD). The first two perform mainly scrambling/descrambling, encoding/decoding, synchronization and throttle the data through the gearbox. Lastly, the PMD consists in the hardware-specific transceiver for the physical medium.

The upper layers are responsible for writing data to the TX FIFO, reading data from the RX FIFO layer and pass it to the user logic.

5.2.1 Ethernet Frame Format

Ethernet systems divide the data to be transmitted in frames in order to exchange data between two endpoints in a network. Each frame contains a source address from the device sending the data and a destination address of the target device. This allows for multiple devices to share the same medium, optimizing the channel capacity. Figure 5.2 illustrates the format of a standard Ethernet frame. The preamble is used for synchronization and contains the pattern 0x55 transmitted 7 times. The Start of Frame Delimiter field marks the start of the frame and contains the pattern 0xD5. The Destination and Source Addresses fields contain the MAC addresses of the initiator and recipient on the network. The Length/Type field is interpreted as the payload length in bytes if its value is lower than 1536, otherwise is considered a type field that may symbolize a VLAN or PAUSE MAC control frame, among others. The next data field is where the payload is carried and can contain up to 1500 bytes in a normal frame or more in special jumbo frames. In case the data field have less than 46 bytes, padding is inserted to ensure a frame length with a minimum of 64 bytes (not including the preamble). Lastly, a frame check sequence (FCS) field is calculated using Cyclic Redundancy Check (CRC) to facilitate error detection by the recipient [Xil16a].

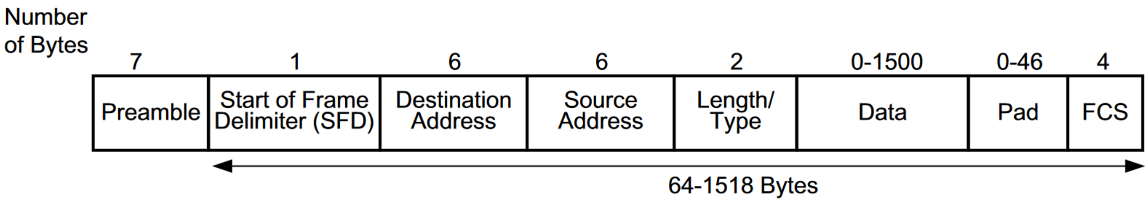


Figure 5.2: Standard Ethernet Frame Format (retrieved from [Xil16a]).

The 10G Ethernet MAC is responsible for inserting the Preamble and Start of Frame Delimiter (SFD), padding if necessary and calculate the Frame Check Sequence (FCS). The remaining fields need to be provided by the user logic.

5.3 FPGA Implementation

The next stage, facilitates the familiarization with development boards based in FPGA similar to the ones used in the original C-RAN test-bed. This next step involves developing a standalone 10 Gigabit Ethernet link over optical fiber to transmit information between two KC705 development kits. Those kits feature a Kintex-7 FPGA along with several connectivity and expansion capabilities. On this particular use case, an emphasis is placed on the SFP cage that allows communication through an optical link using SFP transceivers. Besides that, two FMC connectors (one Low Pin Count (LPC), one HPC) support the addition of daughter boards for additional functionality. A variety of several SMA connectors, USB-to-UART, user push buttons, DIP switches and LEDs complement the board with great flexibility. The development kit with highlighted components is illustrated in Figure 5.3.

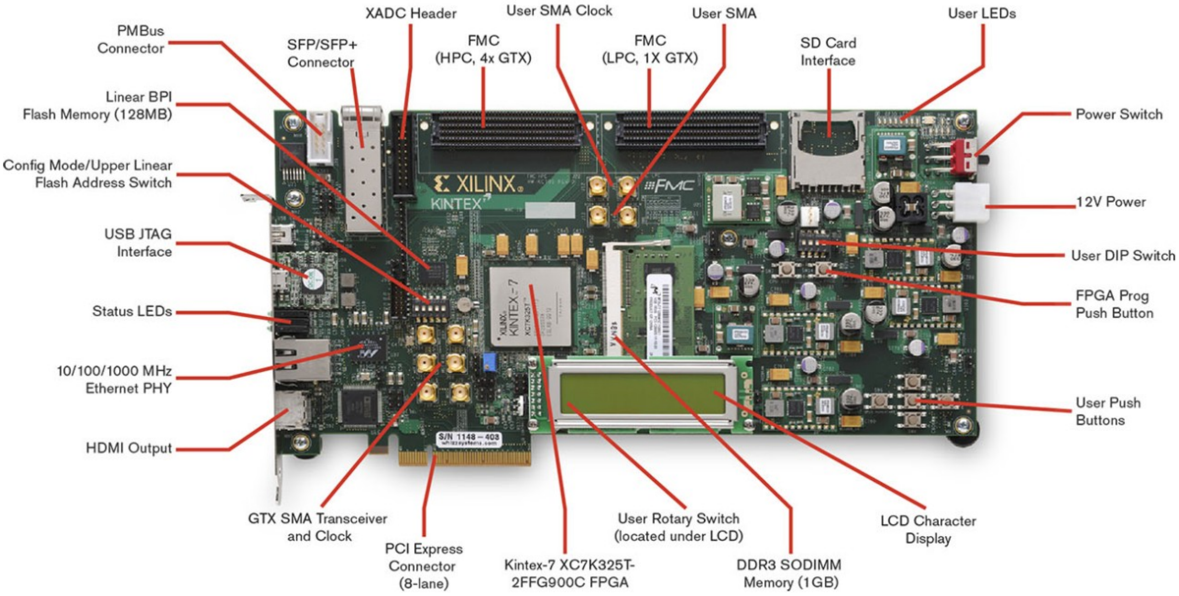


Figure 5.3: KC705 Evaluation Kit (Retrieved from [Xil18b]).

Both hardware and software are developed on Xilinx's development environments - Vivado Design Suite and Software Development Kit (SDK). The Vivado Design Suite allows for hardware development based on a mix of IP cores, block design and hardware description languages such as Verilog or Very High Speed Integrated Circuit Hardware Description Language (VHDL). On the other hand Xilinx SDK design environment allows creation of embedded applications for System-on-a-Chip (SoC) and soft-core microprocessors.

5.3.1 10G MAC and PHY

The Ethernet solution itself is developed around the 10 Gigabit Ethernet Subsystem IP core provided by Xilinx. The subsystem contains the 10 Gigabit Ethernet MAC and 10 Gigabit PCS/PMA connected internally through the XGMII interface. Figure 5.4 represents the 10G Ethernet Subsystem block diagram.

Interfacing both data and management between the user logic and the 10 Gigabit Ethernet MAC are, respectively, AXI4-Stream and AXI4-Lite buses based in the Advanced eXtensible Interface (AXI) of the Advanced Microcontroller Bus Architecture (AMBA) open-standard. Additionally, the subsystem supports high accuracy hardware time-stamping through IEEE1588v2. The 10G PCS/PMA feeds data to the PMD, which in this case is an SFP optical transceiver, through the board Multi-Gigabit Transceivers (MGTs).

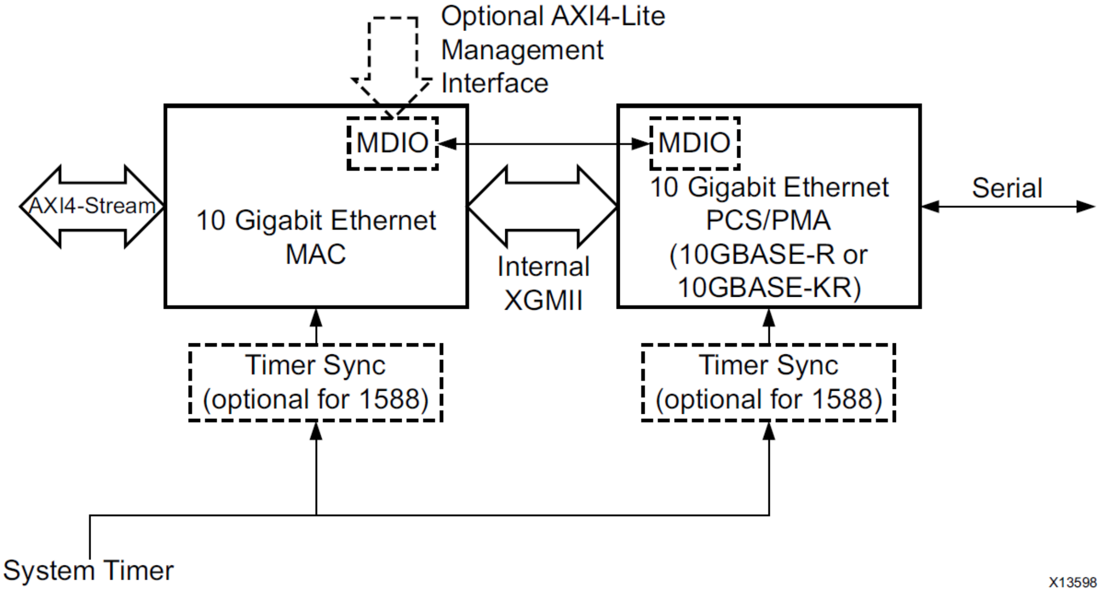


Figure 5.4: 10G Ethernet Subsystem block diagram (retrieved from [Xil16a]).

5.3.2 Multi-Gigabit Transceivers

Before diving into more detail, this section presents the internal Kintex-7 and Virtex-7 FPGAs transceivers architecture in order to provide some background for implementation considerations made later on.

Both KC705 and VC707 use 7 Series GTX transceivers capable of a maximum performance of 12.5 Gb/s [Xil17]. Each board possess several transceivers (16 to the KC705 and 27 for the VC707) providing connectivity with a wide range of serial protocols and line data rates. These transceivers are connected to PCIe, LPC and HPC FMC connectors, SMA connectors, SFP module and the SGMII connection to the Ethernet PHY [Xil16d] [Xil16e].

Each one of these GTX transceivers are arranged in groups of four, forming a GTX Transceiver Quad. Inside each Quad there is a single Quad-based LC tank PLL (QPLL) that together with reference clock selection logic form the GTXE2_COMMON primitive. This primitive is shared between the four GTXE2_CHANNEL primitives, each instantiating a channel-based ring oscillator PLL (CPLL) and the transceiver itself (transmitter + receiver). Figure 5.5 illustrates the composition of a GTX Quad.

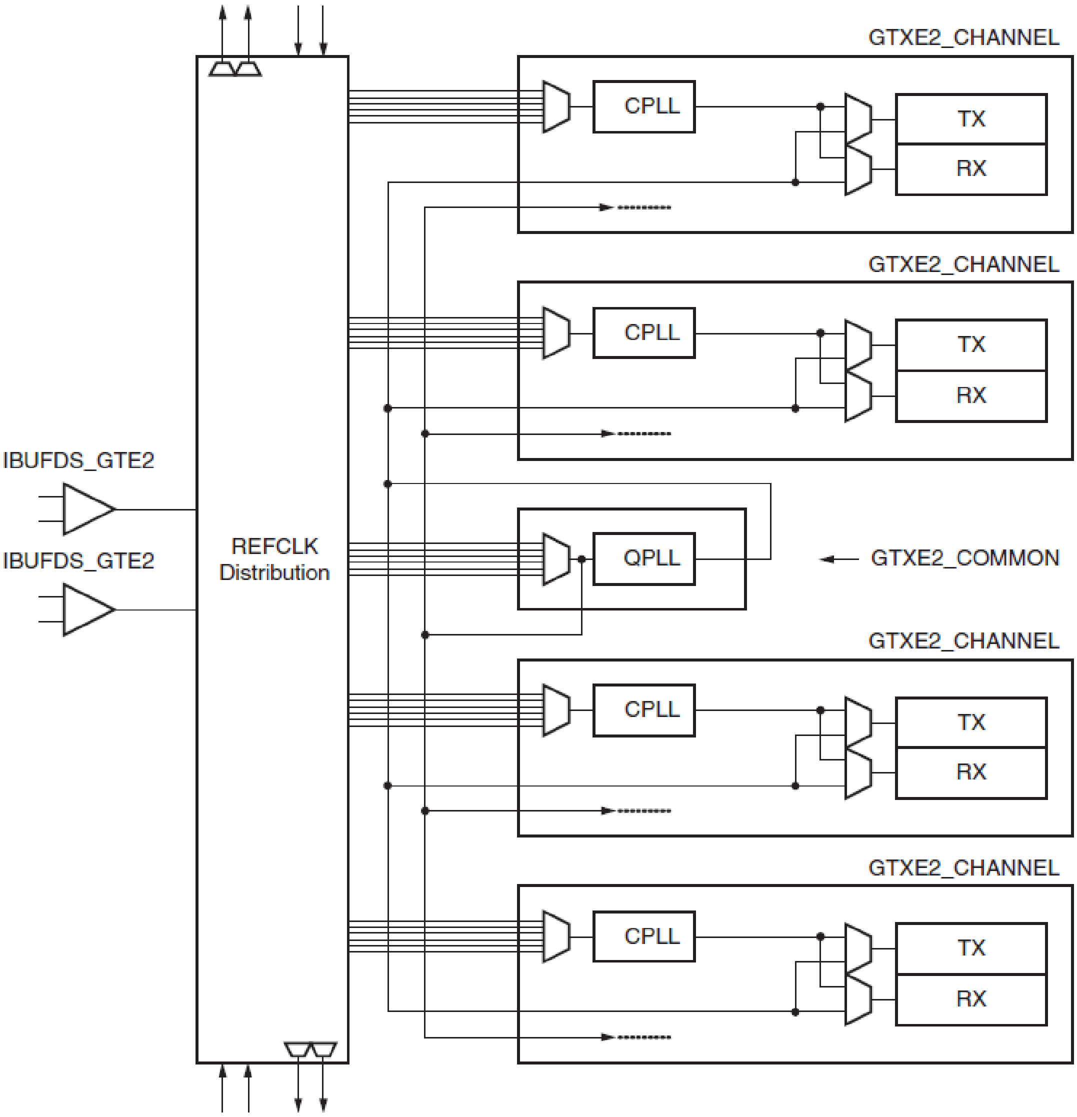


Figure 5.5: 7 Series FPGA GTX Transceivers Quad Configuration (retrieved from [Xil16c]).

Figure 5.6 shows the topology of the transmiter and receiver inside a GTXE2_CHANNEL primitive. Each Quad can then be fed with two pairs of differential reference input clocks or use the routed clocks from the Quad directly above or below. These reference clocks are Quad specific meaning that there is a certain limitation in respective transceiver and reference clock choices. This detail would prove to be a concern at later stages of the integration. Furthermore, the reference clocks are sourced into the FPGA by IBUFDS_GTE2 primitives that act as a input buffer and convert the external differential clocks to single-ended and route it to the REFCLK distribution paths.

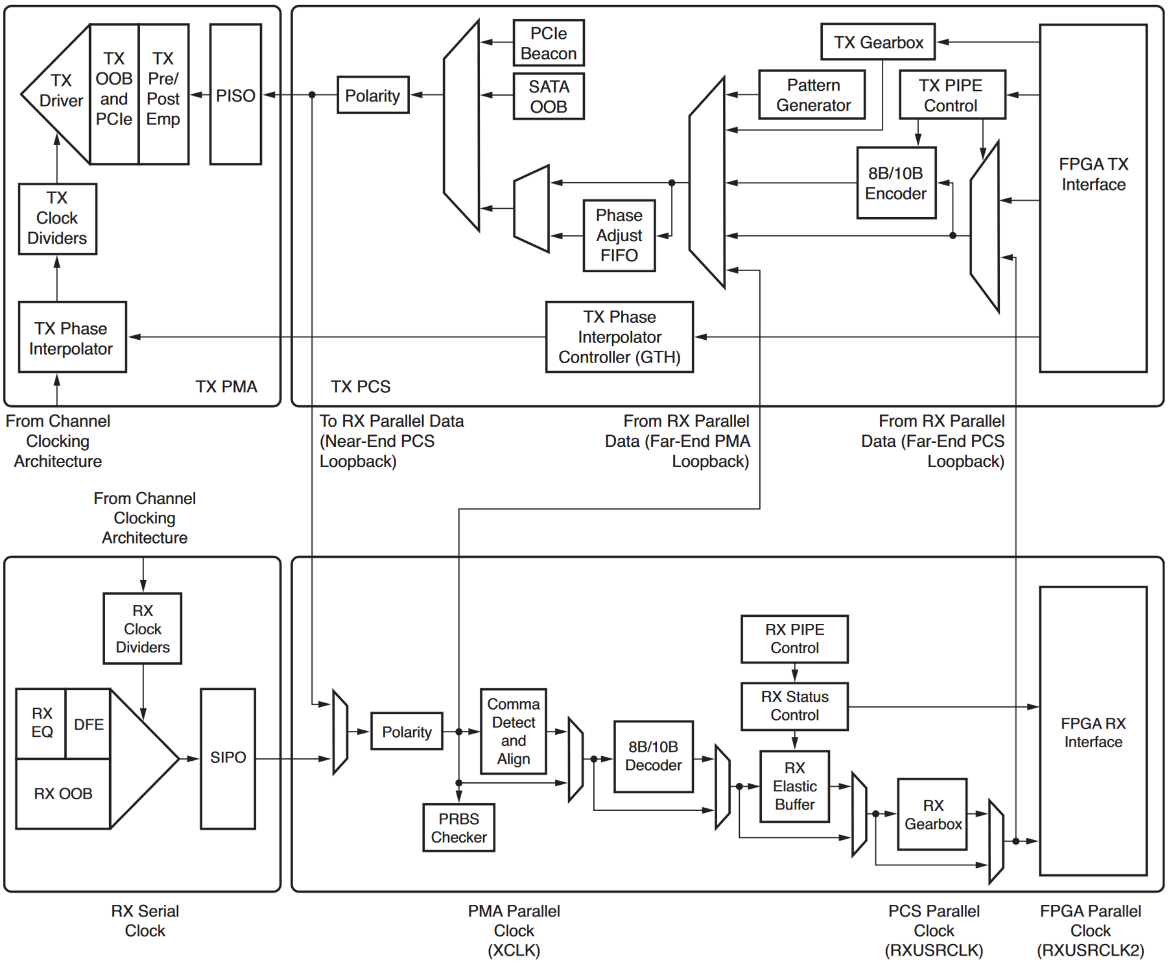


Figure 5.6: GTXE2_CHANNEL primitive topology (retrieved from [Xil16c]).

5.3.3 Example Design

Development of the digital design starts in the Vivado Design Suite v2016.4 by instantiating and parameterizing the previously described 10 Gigabit Ethernet Subsystem IP core through the IP Catalog. The IP core can be parametrized with a AXI4-Stream datapath width of 64-bit clocked at 156.25 MHz or 32-bit clocked at 312.5 MHz. The latter option was preferred as it allows for lower latency [Xil16a].

Vivado also provides a simple example design included with the IP core that is used as a base for the framework. The example design, illustrated in Figure 5.7, besides the 10G Ethernet Subsystem includes:

- TX and RX FIFOs with parameterizable depth size using AXI4-Stream interfaces
- A simple pattern generator and received frame checker
- A control state machine that initializes the system through an AXI-Lite interface
- Clock management logic

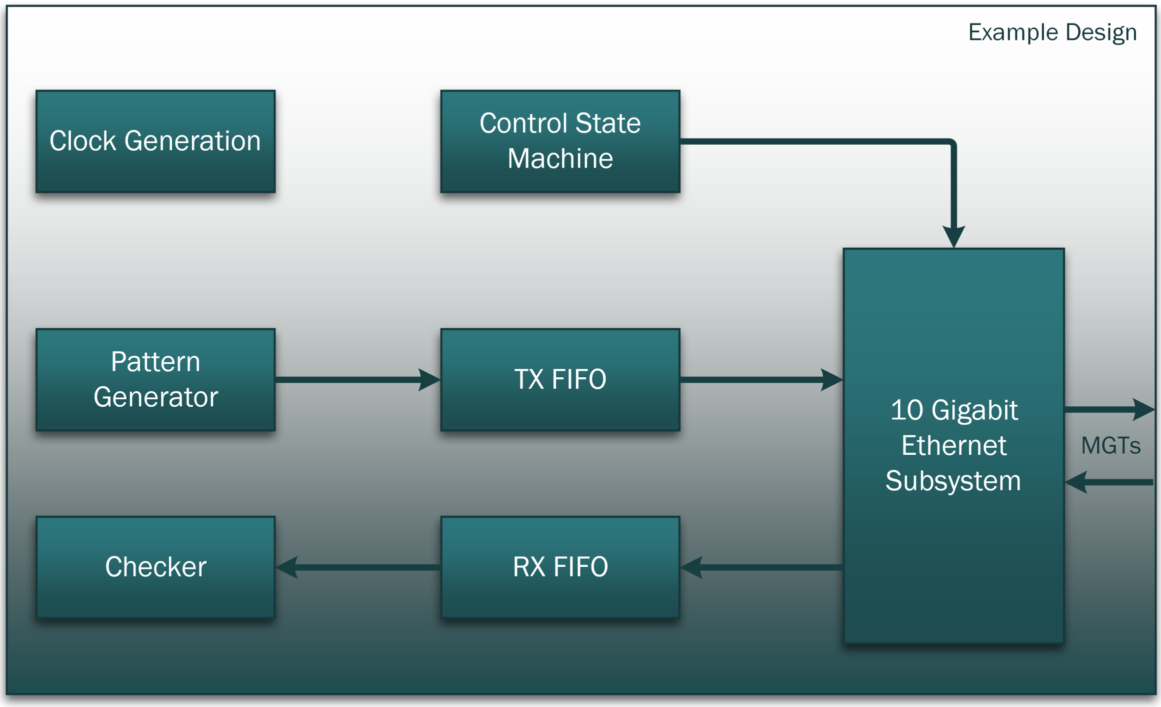


Figure 5.7: AXI 10G Ethernet Example Design (adapted from [Xil16a]).

In a first instance the example design is used "as is" to first ensure correct board and system functionality. The example design support several operation modes that can be configured through input control signals. These modes include [Xil16a]:

- Standard operation: Frames are generated by the pattern generator and transmitted to the TX FIFO, 10G Ethernet System and optical link.
- PCS loopback: Same as before but data is looped back to the receiver at the end of the 10G Ethernet System, before reaching the optical link.
- FIFO loopback: Frames received by the RX FIFO are inserted back to the TX FIFO. In this operation mode source and destination addresses are swapped.

In this case the input control signals were connected to a DIP switch on the KC705 board and can enable/disable the pattern generator, checker, PCS loopback and FIFO loopback. With the exception of the reference clock for the MGTs, the required clocks needed for correct system functionality are generated from the 200 MHz system differential clock of the KC705 through internal Mixed-Mode Clock Managers (MMCMs) or/and PLLs.

At this stage, instead of using the native SFP cage present at the KC705 board, a FM-S14 module was used instead as it already provides the reference clock needed by the MGTs and eliminates the need for an external signal/clock generator.

The FM-S14 is a FMC daughter board that provides up to four SFP module interfaces directly into the KC705 MGTs through a HPC FMC connector [Tec14]. Figure 5.8 shows the FM-S14 module where can be seen the FMC connector at the right side and the four SFP cages on the left.

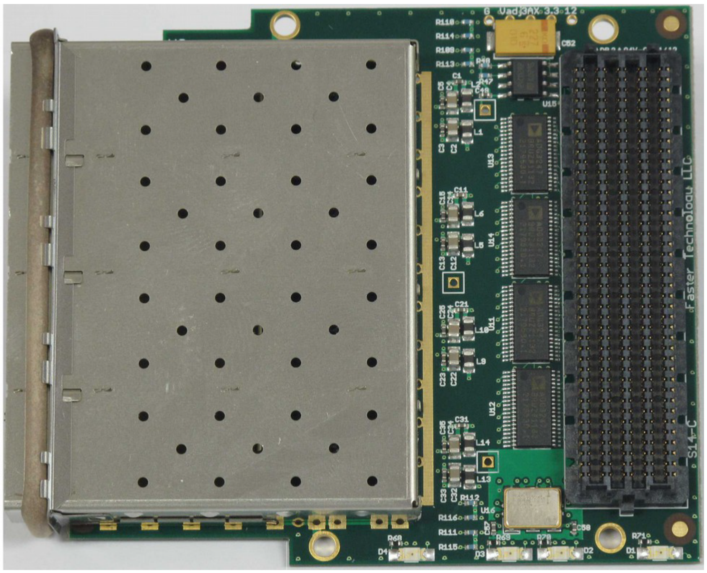


Figure 5.8: FM-S14 - Quad SFP/SFP+ transceiver FMC module (retrieved from [Tec14]).

The FM-S14 includes two programmable reference clocks each providing one of four default frequencies at start-up. Each clock generator frequency is selected by two on-board DIP switches. Default start-up frequency of 312.5 MHz is selected with the DIP switches at position '11'.

For the transceiver choice, a pair of Finisar's FTLX8571D3BCV 1G/10G Dual-Rate SFP+ transceivers were used together with a pair of Multi-Mode Fiber (MMF) patch cord. The FTLX8571D3BCV is a dual-rate transceiver capable of either 1000BASE-SX 1G Ethernet or 10GBASE-SR/SW 10G Ethernet. The connection between SFP transceiver and the board is done through 20 pins as specified in [SFF09]. Table 5.1 shows the most relevant connections as the remaining ones are used for power and ground.

Rate selection is done through pin 7 (RS0) of the SFP: Open or Low enables 1.25 Gb/s; High = enables 9.95 Gb/s to 10.3125 Gb/s [Fin09a]. When using the native SFP cage, the data rate is selected through jumpers J27 and J28 on the KC705. When using the FM-S14 module this is done digitally through the FMC.

Besides the RS0 for rate selection, other pin-out connections necessary for correct system functionality are the TX$_{FAULT}$, TX$_{DISABLE}$ and RX$_{LOS}$. TX$_{DISABLE}$ needs to be tied to ground otherwise laser output is disabled, TX$_{FAULT}$ indicates transmitter fault and RX$_{LOS}$ indicates loss of signal at the receiver. Once again, when using the native SFP cage

Table 5.1: Most important SFP electrical pin-out connections.

Pin #	**Name**	**Description**
2	Tx_Fault	Transmitter Fault
3	Tx_Disable	Laser Output Disabled
4	SDA	Serial Interface Data Line
5	SCL	Serial Interface Clock Line
6	Mod_ABS	Module Absent
7	RS0	Rate Select 0
8	Rx_LOS	Loss Of Signal
9	RS1	Rate Select 1
12	RD-	Receiver Inverted Data Out
13	RD+	Receiver Non-Inverted Data Out
18	TD+	Transmitter Non-Inverted Data In
19	TD-	Transmitter Inverted Data In

TX$_{DISABLE}$ is selected using the KC705 jumper J4 or digitally in case of FM-S14 usage. TX$_{FAULT}$ and RX$_{LOS}$ were connected to the KC705 user LEDs in both use-cases.

5.3.4 Test Setups

Two tests were done with the setup describe previously. Both tests done with standard operation mode, in other words, pattern generator and checker enabled and PCS and FIFO loopback disabled. The first test involves only one KC705 kit, FM-S14 module, one SFP transceiver (FTLX8571D3BCV). A MMF patch cord connects the transmitter and receiver of the SFP module to effectively implement a loopback on the optical link. In this case, frames are generated in the pattern generator, written to the TX FIFO, and read by the 10G Ethernet IP core that sends it to the SFP transceiver through the MGT. At the SFP, data is transmitted over the optical fiber and looped back to the receiver, MGT and 10G Ethernet IP core that writes the data into the RX FIFO. Finally, the frame checker verifies if the received frames were received correctly and checks the pattern on the payload.

The second test is very similar but now involves two KC705 kits (Figure 5.9). One of them making use of the FM-S14 module as described previously and the other using the native SFP cage. In the board using the native SFP cage, a Si570 programmable crystal oscillator (as previously described at **Chapter 4, CPRI based C-RAN Test-bed**) was used to generate the reference clock for the MGT. The generated 312.5 MHz differential clock is routed to the KC705 dedicated SMA MGT reference clock inputs. In this setup, frames are generated by one KC705 board and checked for errors in the other. With the particularity of one of the boards being a Rev 1.0 while the other is a Rev 1.2. Early revisions had the TXP/TXN and

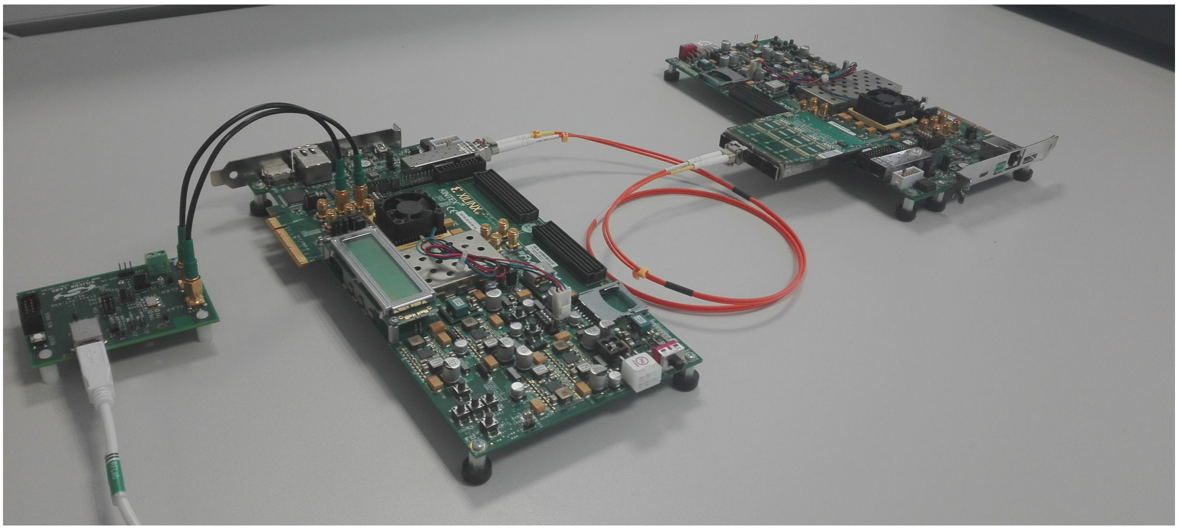


Figure 5.9: 10GE test setup with two KC705, Si570 EVB, FM-S14 and MMF Patch Cord

RXP/RXN polarity inverted as described in [Xil14a]. In order for Rev 1.0 boards to work successfully with Rev 1.1/1.2 boards the FPGA MGTs TX and RX polarity control attributes need to be adjusted.

Functionality is confirmed through the KC705 user LEDs. They display pattern generator/checker activity, error detection by the checker and MGT status. The setup also allow deliberated insertion of errors and reset of the frame error LED through on-board user push-buttons.

5.4 Digital Design

After basic functionality successfully tested the next step is to perform the necessary modifications on the example design to allow the transport of IQ radio samples over Ethernet. As such, the pattern generator, checker, address swap, loopback and complementary logic are removed from the design. The TX/RX FIFOs, control state-machine and most of the clock management logic are maintained. For the rest of this document, the TX/RX FIFOs mentioned previously are considered an integral part of the 10G Ethernet Subsystem core while future mentions are part of the interfacing logic between the 10G Ethernet solution and other parts of the design. Next subsections explain the logic blocks developed for this use-case. First a "Framer" is needed to frame the IQ radio samples into the format required by the 10GE MAC. A "De-Framer" does the opposite job of extracting the IQ radio samples from the frame payload received from the 10GE MAC. A pair of FIFOs are also needed to interface the logic generating/consuming the IQ radio samples and the Framer/De-Framer. Figure 5.10 illustrates the concept:

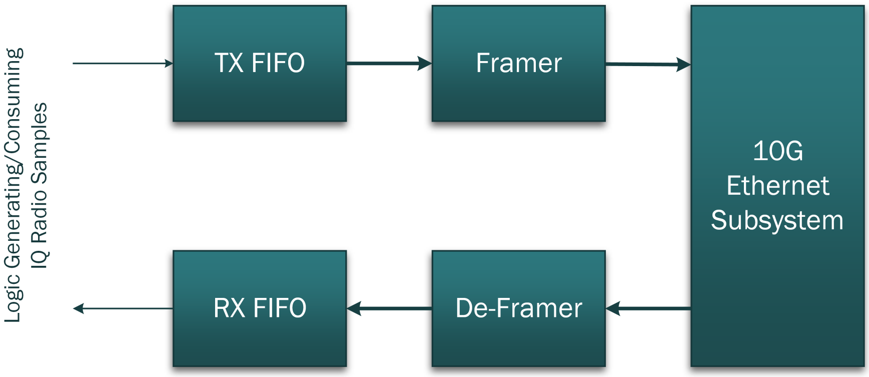


Figure 5.10: Interfacing blocks for the 10G Ethernet Subsystem.

5.4.1 Framer/De-Framer

The Framer block frames the incoming data as required by the 10G Ethernet MAC using a state machine implementation. The Framer starts by checking if the transmission FIFO has enough data for an entire frame and in case the MAC is ready starts to build the frame according to specifications in Figure 5.11. T$_{VALID}$ is asserted and data is provided to the MAC. In the first clock cycle are transmitted the destination address and the first 16 bits of the source address. In the second clock cycle the remainder of the source address is sent along with the Length/Type field. There's also 2 bytes of data that can be transmitted in this clock cycle. For the sake of keeping the data aligned within each clock cycle these 2 data bytes available are used as a frame counter to allow easy monitoring of received frames on the receiving side. On the remaining clock cycles the payload is transmitted.

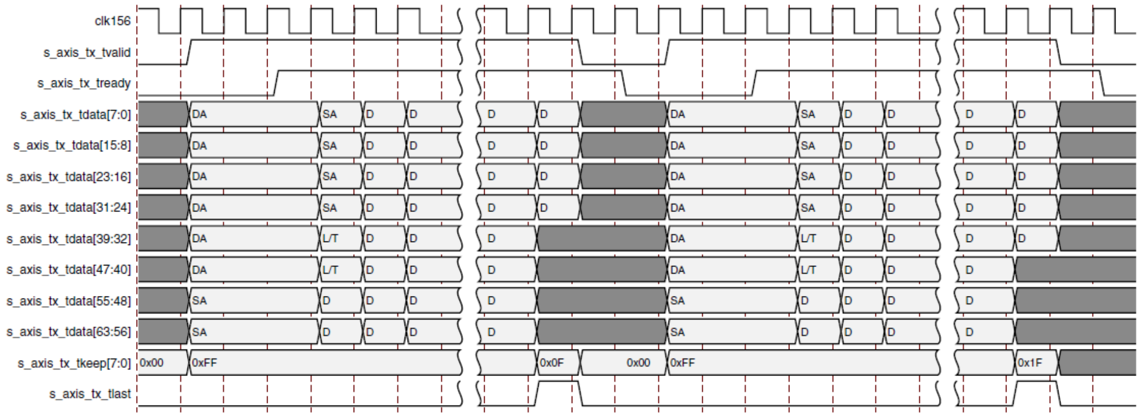


Figure 5.11: Standard Frame Transmission (retrieved from [Xil16a]).

At this stage and since the direct transport of IQ radio samples allows for great flexibility, the number of samples per frame need to be predetermined. As a compromise between having enough samples to not waste bandwidth with padding and keeping the frames relatively short for lower latency, it was decided to build the frame with a payload of 64 bytes. As such, the next 8 clock cycles are fully dedicated to the payload (in this case, IQ radio samples). In the last clock cycle the flag T_{LAST} is asserted to signal the end of frame to the MAC. Flag T_{KEEP} is set appropriately if the packet ends at a non-64-bit boundary [Xil16a]. Frame transmission only occurs if T_{READY} (controlled by the MAC) is asserted. Otherwise transmission is paused until T_{READY} is asserted again.

The De-Framer implements analogous functionality to the Framer (also implemented through a state-machine), accordingly to Figure 5.12. In order to synchronize with incoming frames, the logic waits for T_{LAST} to be asserted. In the next cycle, if T_{VALID} is asserted, frame reception starts. MAC address fields and payload length is verified and an error flag is generated by computing the frame counter. The payload is extracted and provided to the user logic. Frames received with CRC errors are signaled by the core by keeping T_{USER} deasserted while asserting T_{LAST} at the end of the frame.

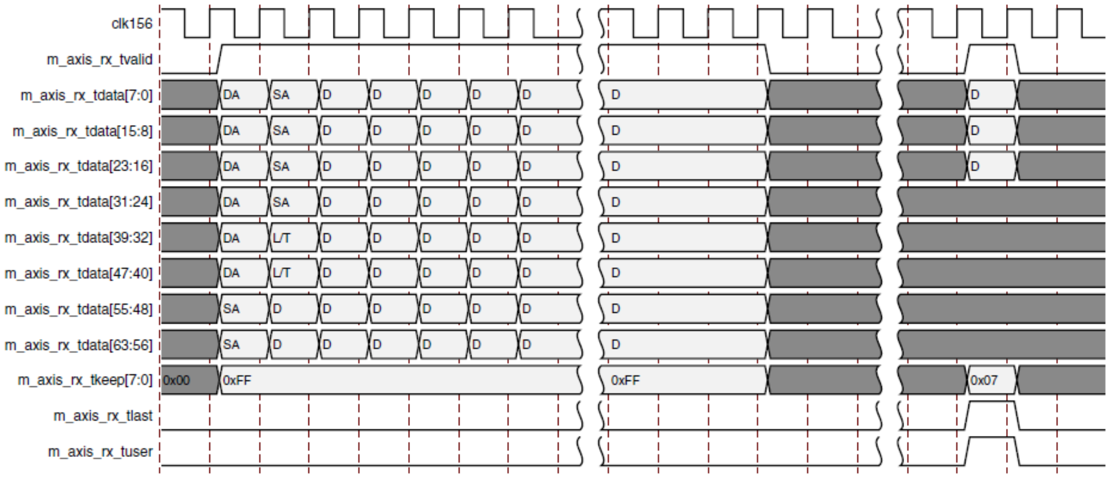


Figure 5.12: Standard Frame Reception (retrieved from [Xil16a]).

Both Framer and De-Framer were developed entirely in Verilog hardware description language and its validation performed in simulation environment in Vivado. Figure 5.13 exemplifies the validation process of the "Framer" module. Analysis is as follows:

1. Reset applied;
2. FIFO programmable flag *prog_empty* goes low, indicating that there is enough words for the construction of one frame (8 x 64 bits);
3. As a result, the framer starts building the frame by providing the DA (a1111111111b) and the SA least significant bits (222d) to the MAC (port *data_out*) and assert *tvalid*;
4. The Framer holds the data in *data_out* until *tready* is asserted;
5. In the next clock cycle, SA remaining bits (0xc2222222), length field (0x0040) and two bytes of data (frame number counter - 0x0001) are provided in *data_out*. At the same time, *read_enable* is asserted to indicate the FIFO to provide data in the next clock cycle;
6. For the next 8 cycles, data read from the FIFO (*data_in*) is provided to the MAC through *data_out*;

7. In the last cycle, *tlast* is asserted to signal the MAC that it is the end of the frame;
8. Building of the next frame begins again.

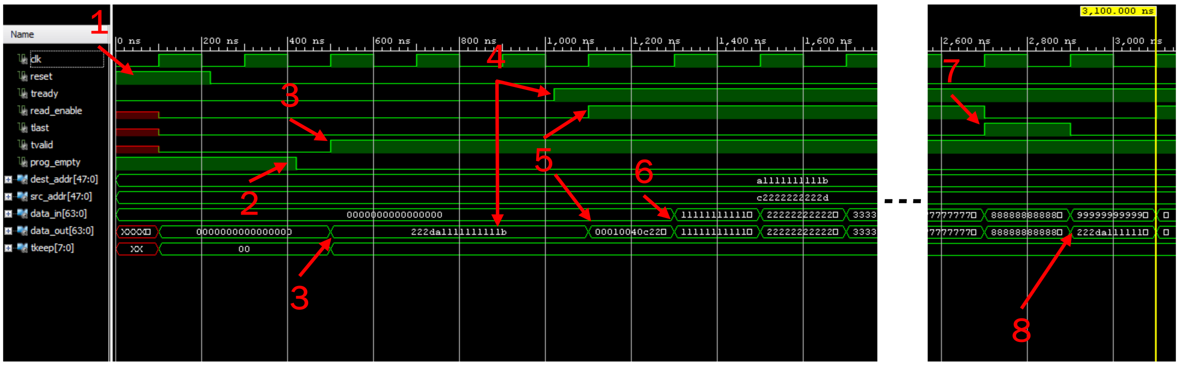
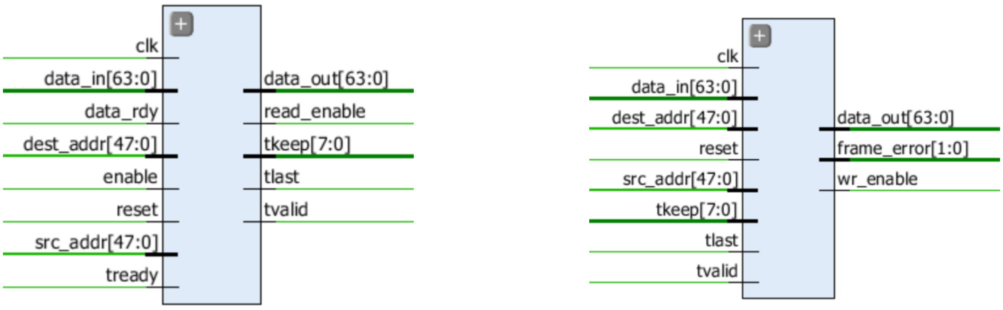


Figure 5.13: "Framer" module validation through behavioral simulation in Vivado.

Similar simulations were performed to evaluate correct functionality of the De-Framer. The simulations were based in testbenchs also written in Verilog to cover several possible scenarios. Figure 5.14 illustrates the inputs and outputs of the final Framer and De-Framer logical blocks.



(a) Framer

(b) De-Framer

Figure 5.14: Framer and De-Framer logical blocks.

5.4.2 FIFOs

The TX and RX FIFOs are implemented using the FIFO Generator core from the Vivado IP Catalog. They are based in Block RAMs and feature individual read/write clocks and data widths in order to interface the Framer/De-Framer (10GE logic) with the logic providing the IQ radio samples. The TX FIFO is configured to have a 32 bit write data word and a 64 bit read data word while the RX FIFO has symmetric parameters. Both FIFOs have status flags in the form of programmable threshold empty and full flags ensuring that the FIFOs are only read when there is at least one complete frame and written when there is enough space for a complete frame.

5.5 Test Setup

To test the correct operation between Framer/De-Framer and the 10G Ethernet core it was used the setup illustrated on Figure 5.15. A simple 32 bit counter emulates the IQ radio samples to transmit. The counter writes to the TX FIFO and when there are enough words the framer starts reading the FIFO, builds the frame and transmit it to the MAC. On the receiving side, the de-framer verifies received frames and if everything is correct and there is available space, writes the payload to the RX FIFO. A simple checker verifies if received data is composed by incremental words and if not, signals an error flag by switching on a user LED in the KC705 board. This LED can be reset by pressing a push-button.

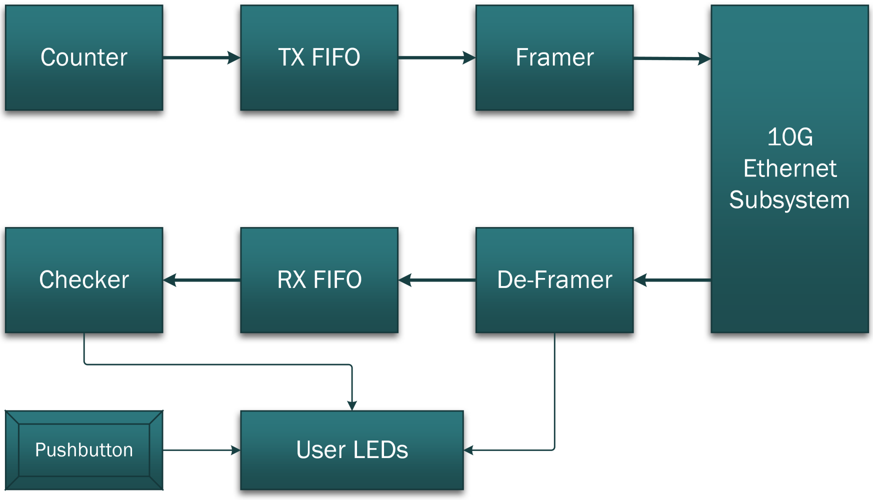


Figure 5.15: Standalone 10G Ethernet test block diagram

Development of previous logic blocks relied significantly on the simulation tools available in the Vivado Design Suite, including behavioral and timing simulations making use of testbenchs to cover several scenarios. On-board functionality is then verified by a mix of user LEDs and ILA Debug cores explained in the next section.

5.6 ILA Debug Cores

Although not required for the normal system operation, debug cores were an essential part of the project development as it allows to verify the design functionality by debugging on-board FPGA signals. Debugging is performed using Vivado Integrated Logic Analyzer (ILA) cores. This feature allows in-system signal monitoring by implementing additional logic onto the FPGA that captures data based in hardware triggers or user inputs at system speeds. Data is stored in Block RAMs and sent to the computer via USB cable were the Vivado logic analyzer is used to interact with it [Xil16f]. Multiple ILAs debug cores can be created, with each one managing the probes within the same clock domain. Each ILA have two types of input ports. Various probe ports connected to the signals intended for debugging and a single clock input port used to register the probe values. It is recommended to use the same clock signal that is synchronous to the design logic that is attached to the probe ports of the ILA core [Xil14b]. The sample data depth, representing the maximum number of samples that can be stored at run time for each probe, can be configured when instantiating the ILA core and can range from 1024 to 131072.

Although undoubtedly useful, these debug cores are also another source of potential problems. Since the debug cores are implemented using additional FPGA resources they can affect the original hardware since the synthesis and implementation Vivado tools need to place and route more logic, potentially leading to timing violations e.g., worst and total negative slack. This can lead to malfunction of the original hardware intended to be debugged or faulty reading of data from the ILAs debug cores. This aspect would turn out to be particularly problematic on later integration stages on the BBU side FPGA since it is already heavily populated.

After achieving a working 10 Gigabit Ethernet solution, the next chapter describes the integration of said solution with the original C-RAN test-bed.

CHAPTER 6

Ethernet-based C-RAN Test-bed

Integration of the new fronthaul implementation on top of the existent C-RAN laboratory test-bed.

6.1 Introduction

This chapter describes the integration of the 10 Gigabit Ethernet solution developed in **Chapter 5, 10 Gigabit Ethernet Link Implementation** with the C-RAN laboratory test-bed presented in **Chapter 4, CPRI based C-RAN Test-bed**.

The same KC705 evaluation board equipped with a Kintex-7 FPGA and VC707 evaluation board featuring a Virtex-7 FPGA provide the foundations for the new fronthaul interface implementation.

Since the original digital design (already presented in **Chapter 4, CPRI based C-RAN Test-bed** as the underlying base for this dissertation) is fairly complex, the initial strategy is to perform minimal modifications in order to first guarantee basic functionality and then work towards better performance and efficiency. As such, while the CPRI core is replaced by a 10Gbit Ethernet solution, the EUTRA modules are kept since they already provide the interface to the Block RAMs used to interchange data between the computer and VC707 development board. Additionally in this first iteration, data, management and synchronization that were previously all part of the same CPRI link are now treated separately. Data is transmitted through the 10 Gigabit Ethernet optical link. Control and management information are exchanged over a Gigabit Ethernet connection over a standard ethernet copper cable (RJ45). Finally the necessary system clocks are generated locally at each development board.

It is clear that this would not be feasible in a real world scenario but it is important to emphasize that this dissertation's objective is to provide a working proof of concept and that further development and optimization will be required. In the future, control plane data could share the 10 Gigabit Ethernet link with user plane data and synchronization achieved with the support of Synchronous Ethernet (SyncE) and/or IEEE 1588v2 Precision Time Protocol.

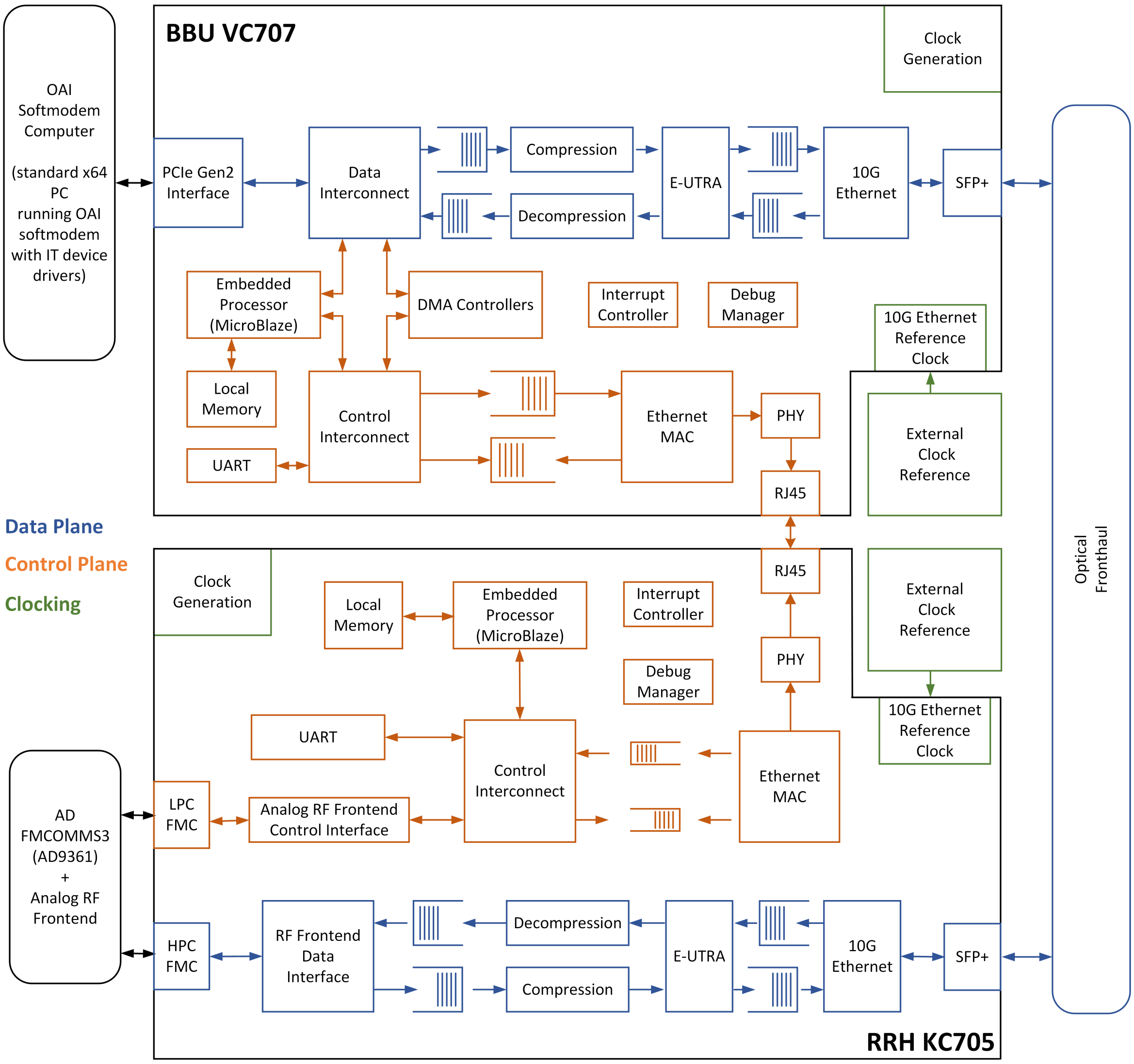


Figure 6.1: Global digital design system architecture

Figure 6.1 illustrates the global digital design. The previous CPRI core is now replaced by the 10 Gigabit Ethernet solution developed in the previous chapter. Note that, since it is planned to keep the EUTRA modules, additional interfacing logic (presented in following sections) is required in order to adapt the EUTRA and 10GE interfaces. The control and management links that were previously connected to the CPRI core are now routed to the on-board 10/100/1000 Mbit/s tri-speed Ethernet PHY and subsequently RJ45 connector. The clock recovery logic in the RRH is now unnecessary since it has all necessary clocks locally generated.

Integration begins by making the necessary changes in the original digital design in the Vivado Design Suite. Development was done on an older Vivado version (v2014.4) since the original project was developed with this version and an update to more recent Vivado versions would necessarily require the upgrade of every IP core in the original project. This caused system malfunction since the new revision of some IP cores had new or altered functionality and input/output ports.

6.2 Data Link

The 10Gbit Ethernet solution now replaces the old CPRI core to transfer the user data plane (digitized IQ radio samples). First the CPRI IP core is removed along with associated logic. As mentioned before, the EUTRA modules are kept and the 10G Ethernet solution developed previously is instantiated. As a side note, Vivado v2014.4 10G Ethernet Subsystem IP core v2.0 did not yet support the 32-bit datapath width option as specified on the IP core revision history at [Xil16a]. As such, the IP core needed to be re-parameterized to make use of the 64-bit datapath width. Necessary changes were made and the reference clock to the MGT was altered to 156.25 MHz as required.

In order for the new 10 Gigabit Ethernet core to communicate successfully with the EUTRA module, additional interfacing logic is required.

6.2.1 Interfacing with the EUTRA Core

Since the 10G Ethernet core (and Framer/De-Framer) operates with a 64-bit datapath while the EUTRA modules have a 32-bit width data bus, the next step is to implement a block to convert the data width. This is accomplished by FIFOs with different read and write data widths. Two of these FIFOs are implemented: one for transmission and another for reception. Additionally, the FIFOs also provide independent read and write clocks acting as a clock domain crossing as data is generated and consumed at a lower rate than it is being transmitted. Finally, the "Data Interface Controller" logical block is needed in order to generate some flags required by the EUTRA modules that were previously generated by the CPRI core. An overall view of these support modules is presented in Figure 6.2. The figure also illustrates the two distinct clock domains.

The following subsections provide more detail in each of the logical blocks mentioned above. The Framer and De-Framer are the same modules developed in the previous chapter. With the exception of the FIFOs instantiated using the Vivado IP catalog, each block was developed in Verilog and tested, individually at first and collectively later, using behavioral and timing simulations as needed in order to ensure correct functionality.

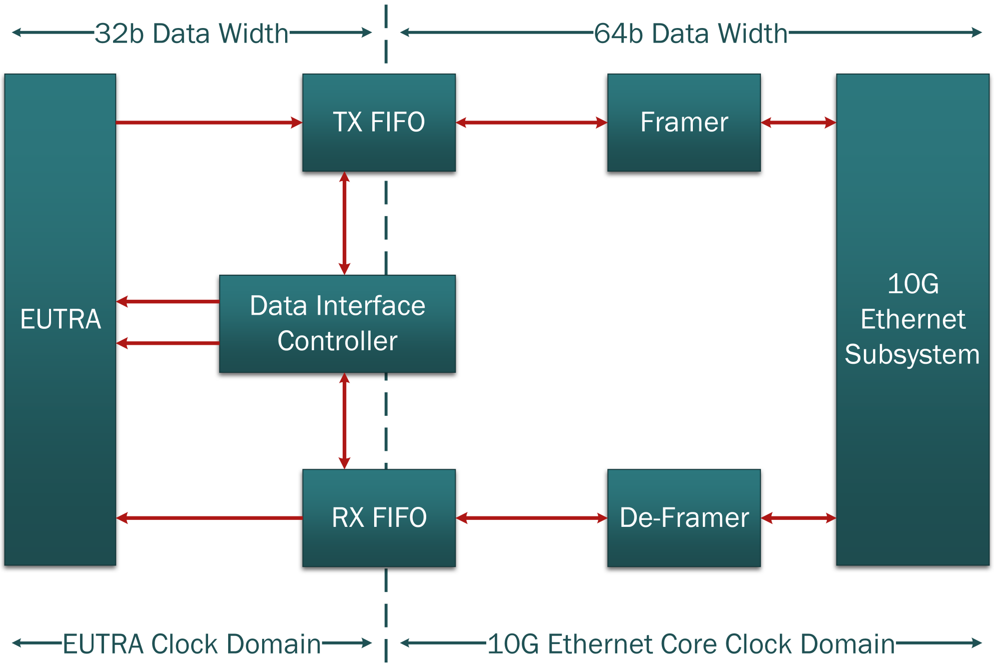


Figure 6.2: Logical blocks interfacing the EUTRA modules and 10G Ethernet Core

6.2.2 FIFOs

The TX and RX FIFOs are implemented using the FIFO Generator core from the Vivado IP Catalog [Xil15]. They are based in Block RAMs and feature individual read/write clocks and data widths. The TX FIFO is configured to have a 32 bit write data word (from the EUTRA module) and a 64 bit read data word (to the Framer) while the RX FIFO has symmetric parameters. Both FIFOs have status flags in the form of programmable threshold empty and full flags ensuring that the FIFOs are only read when there is at least one complete frame and written when there is enough space for a complete frame. The FIFOs use a clock of 30.72 MHz for the EUTRA side (TX FIFO write clock and RX FIFO read clock) and a 156.25 MHz clock for the 10G Ethernet side (TX FIFO read clock and RX FIFO write clock). Both FIFOs feature a depth of 4098 64-bit words (or 8192 32-bit words).

6.2.3 Data Interface Controller

The last block is designated Data Interface Controller and manage the data exchange between the EUTRA module and the TX/RX FIFOs by generating all necessary flags to the EUTRA module, previously originated in the CPRI core. These flags are the *basic_frame_first_word*, asserted at the start of a new basic frame being received from the 10G Ethernet core and *iq_tx_enable*, signaling the EUTRA to start transmitting the next basic frame to be sent over the fronthaul. Required timings are illustrated in Figure 6.3 as specified by the CPRI Specification [AB+15].

Although in Figure 6.3 IQ data is represented as a 16-bit wide, the EUTRA module is in fact implemented with a 32-bit wide bus for the *iq_rx* and *iq_tx*. As a result there is only

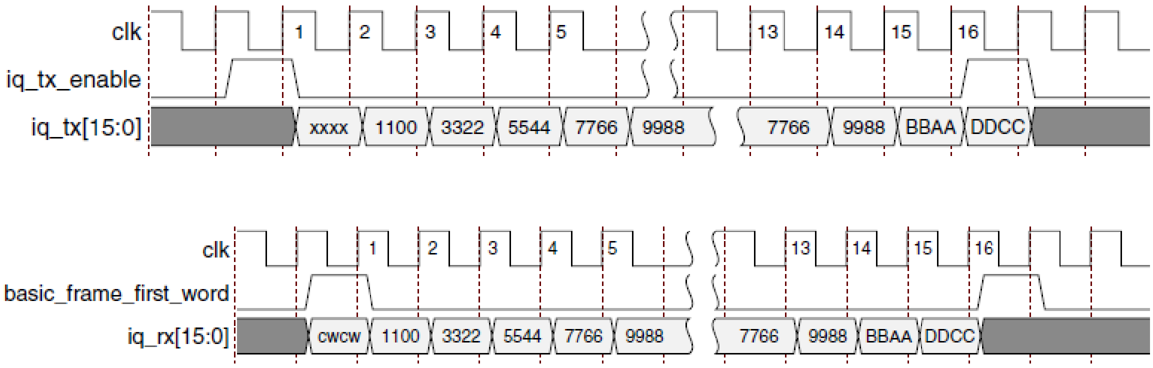


Figure 6.3: Timings required for the EUTRA flags (retrieved from [AB+15]).

the need for 8 clock cycles instead of the 16 represented for transmission/reception of a basic frame.

It also controls the TX FIFO's write enable flag and RX FIFO's read enable flag to ensure synchronism with the *basic_frame_first_word* and *iq_tx_enable* flags.

6.3 Control Link

For the control and management link, it's used the same Ethernet MAC as in the original project (based in the AXI Ethernet Subsystem IP core from Xilinx [Xil16b]) but instead of connecting to the CPRI core, is now linked with the 10/100/1000 Mbit/s tri-speed Ethernet PHY present in the VC707 and KC705 boards. The PHY then connects directly with the RJ45 Ethernet jack.

As mentioned before in Section 5.2, connecting the PHY and MAC is the GMII interface or one of it's variants. On the BBU VC707 this would turn into a conflicting problem involving the MGTs as the VC707 only supports Serial Gigabit Media-Independent Interface (SGMII) and therefore needs an MGT for serial transmission at 1 Gb/s. As explained previously in Section 5.3.2, 7-Series MGTs are arranged in banks of four (forming a GTX transceiver Quad) and share some clocking logic between them: reference clocks, QPLL, etc. Both SFP and SGMII GTX transceivers are located on the same transceiver bank (Quad) and therefore need to share the same GTXE2_COMMON primitive. Since both 10G and 1G core subsystems instantiate its own GTXE2_COMMON while sharing the same GTX Quad leads to errors during implementation. To complicate matters even further, the 10G subsystem has a effective line rate of 10.315 Gbps and needs a reference clock of 156.25 MHz (from the dedicated SMA MGT REFCLK inputs) and the 1G Ethernet Subsystem has a line rate of 1.25 Gbps and a reference clock of 125 MHz (from the "SGMII GTX Transceiver Clock Generation" on the board).

Although this problem could be probably solved with available logic in the FPGAs MGTs by meddling with each transceiver individual CPLL, modifying clock dividers, and distinct reference clocks sources, it would not be a trivial task. A simpler solution that was ultimately adopted was to use the FM-S14 module (used in **Chapter 5, 10 Gigabit Ethernet Link Implementation**) instead of the VC707 native SFP cage as the MGTs for the FMC connectors are in a separate GTX transceiver bank (Quad).

This was not a problem on the KC705 at the RRH side since unlike the VC707, it supports the GMII, RGMII and SGMII to interface the Ethernet PHY and MAC. As such either GMII or RGMII can be used as they are parallel interfaces and don't required the use of a MGT for serial transmission.

6.4 Clocking

Description of the clocking aspect of the whole setup was intentionally left for last as it turned out to be much more challenging than initially though and the source of several setbacks. Lets first describe the main clock domains and how/where they are generated.

6.4.1 MGTs Reference Clock

First, the MGTs reference clocks. This clock is used by the MGTs and for all user logic related to the 10G Ethernet side. At the BBU VC707, since the FM-S14 module is used instead of the native SFP cage, the included programmable oscillator is used. However, the reference clock frequency of 156.25 MHz that is now needed (since we're using the 64-bit datapath for the 10G Ethernet IP core) does not match any of the possible four default start-up frequencies. Although the oscillator could be programmed through a I^{2}C interface with any frequency between 15.48 MHz and 1300 MHz, a simpler way to achieve the same reference clock of 156.25 MHz is to activate the clock divider (/2) signal on the MGT logic on the FPGA while keeping the default start-up frequency of 312.5 MHz in the FM-S14. On the other hand, the RRH KC705 uses the native SFP cage and a Si570 programmable crystal oscillator in a external evaluation board. The Si570 does not need to be programmed as in Section 4.4.1 since it's default start-up frequency is exactly the required 156.25 MHz.

6.4.2 EUTRA Clock Domain

The other major clock domain is the "EUTRA clock" at 30.72 MHz. This clock was on a first instance generated locally at each board, using MMCM primitives implemented through the Clock Wizard IP core from Vivado IP Catalog. The input clock for these MMCMs were the 200 MHz differential system clock present at each board.

6.4.3 AD9361 Reference Clock

On the RRH side, there is also the need for a 61.44 MHz reference clock for the AD9361 RF transceiver. On a first instance, a 15.36 MHz clock was also derived from the same MMCM as the 30.72 MHz clock mentioned previously. This clock signal was then routed outside of the FPGA to the CDCE72010EVM external low jitter clock synchronizer through a SMA cable before being redirected as the 61.44 MHz reference clock to the AD9361.

In this first instance, the way this clock signals are generated and where they are sourced from would later prove to be the cause of the inability for the User Equipment (UE) to correctly connect and attached to the C-RAN. More of this matter in the next chapter.

6.5 Fronthaul Infrastructure

Regarding SFP choice, neither the Finisar's FWLF-1631-xx used in the original CPRI setup nor the FTLX8571D3BCV used for the 10G Ethernet standalone implementation fulfill the requirements needed for this use-case. The first pair only support maximum bit-rates of 2.7 Gbps and therefore unfit for 10 Gbps Ethernet while the FTLX8571D3BCV while supporting 10G Ethernet is only capable of reaching 400 meters using OM4 multi-mode fibers and consequently can't support the 20 km fronthaul.

As a result, a pair of FTLX1672D3BTL, also from Finisar, were used instead. The FTLX1672D3BTL is a multi-rate SFP transceiver capable of links up to 40 km over single mode fiber and bit-rates ranging from 1.2 Gbps to 11.3 Gbps by using a 1550 nm laser. It isn't however DWDM compliant and as such can't be used in conjunction with the DWDM C-band AWG used in the original test-bed. For this reason the AWG was replaced by another optical circulator. This way bi-directional transmission over a single fiber can be achieved.

6.6 Embedded Application

The embedded application running on top of the MicroBlaze soft-core processor (already described in Section 4.6) also needs to be adjusted to account for the modifications on the digital design.

In this case only small modifications need to done regarding the CPRI core initialization and configuration. These functions are now longer needed and as such are removed from the project. Remaining functions are kept as is. The only new addition to the embedded application is the insertion of messages through the UART to help overall system debugging.

The next chapter presents the validation of the new fronthaul link, optimizations required and results.

CHAPTER 7

Validation, Optimization and Results

Ethernet-based C-RAN testbed validation, optimization, tests and results.

7.1 Introduction

With the digital design bitstream successfully generated for each FPGA without any timing constraints violations, changes on the SDK completed and required modifications in the hardware infrastructure done, the first iteration of the integration described in the previous chapter is now completed. Setup validation follows next using a Commercial Off-The-Shelf (COTS) smartphone (Samsung Galaxy S4 Active) as User Equipment (UE).

7.2 Validation and Optimization

The first thing to be done was to verify that data was being transmitted successfully from BBU to RRH and vice-versa, respecting the required timings and verify the presence of errors in the fronthaul. For this, the external Si570 clock generator and CDCE72010EVM low jitter clock synchronizer are initialized and programmed. Both FPGAs are turned on and programmed with the new bitstreams. Initial validation consists in analyzing the required signals through the ILA debug cores on both boards.

Figure 7.1 highlights the most important signals like the *basic_frame_first_word* and *iq_tx_enable*, crucial to the correct exchange of data between the EUTRA and 10GE cores. Another signal carefully monitored was the *fn_error*, this flag is asserted when the De-Framer block detects a missing or corrupted frame. The top ILA shows signals from the 10GE clock domain and the bottom ILA shows signals from the EUTRA clock domain. The start of a new basic frame is easily spotted because the last 4 clock cycles must be 0x0 since the EUTRA

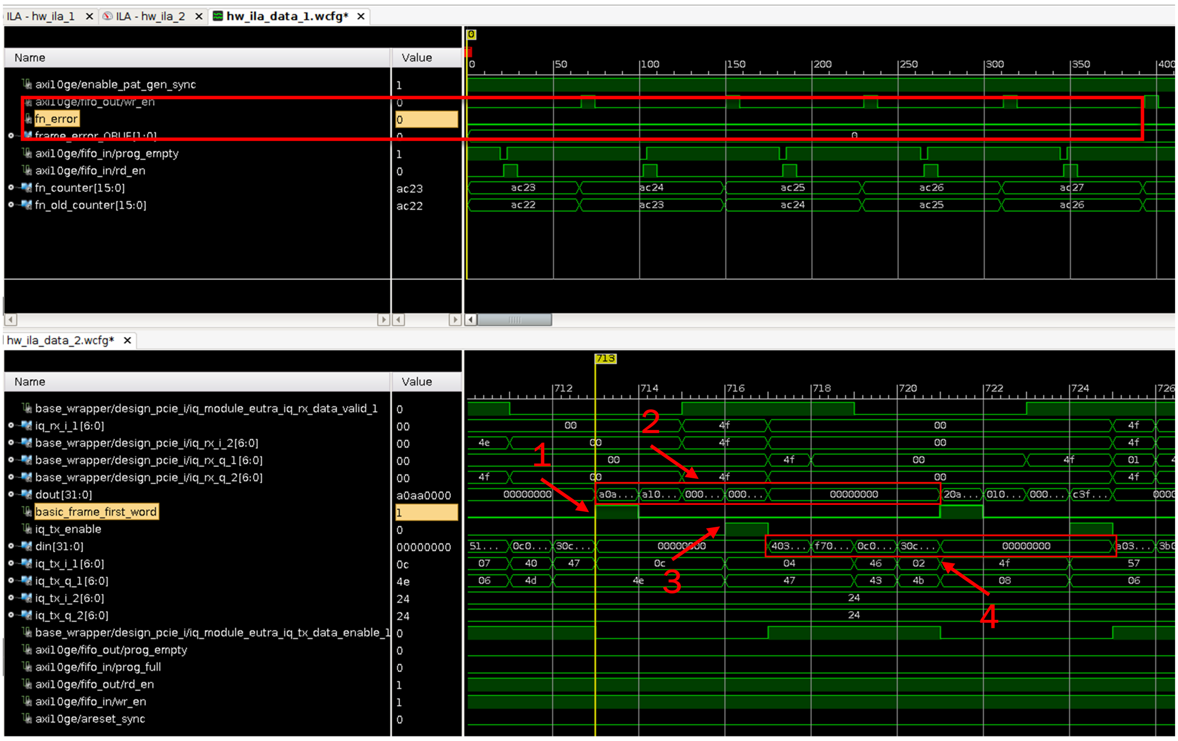


Figure 7.1: Validation of the BBU design signal through the ILA.

module was originally implemented with two channels, but only one is being used. Figure 7.1 analysis of most important signals:

1. *basic_frame_first_word* goes high;
2. New basic frame reception starts in the same clock cycle, being transmitted from the RX FIFO to the EUTRA core (*dout*).
3. *iq_tx_enable* goes high;
4. Transmission of a new basic frame starts in the next clock cycle, being transmitted from the EUTRA core to the TX FIFO (*din*).

Although no BER test was conducted, the setup was left running for several hours with an ILA trigger ready on the *fn_error* flag to check for lost frames. After several hours running, every frame transmitted was successfully delivered on both directions. Validation results from the RRH KC705 are identical.

7.2.1 Synchronization Problems

The first adversity emerged when booting up the OpenAirCN (EPC) and running the OAI LTE soft-modem (BBU). The LTE soft-modem would sporadically crash and while it was running the smartphone could not find the network. When performing a mobile network search, it could only detect current commercial mobile network operators (MEO, Vodafone and NOS). The first thing to do was to verify if the AD9361 RF transceiver was emitting the correct spectrum with a Rohde&Schwarz FSP Spectrum Analyzer.

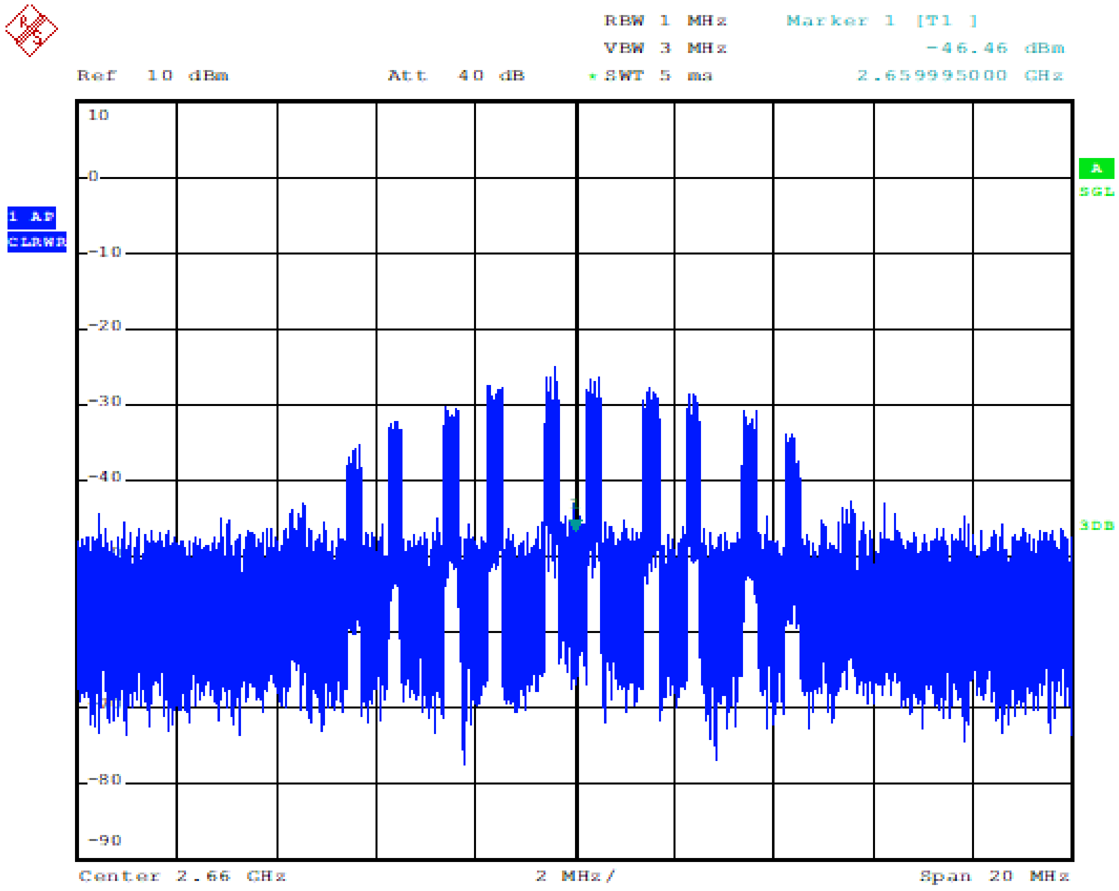
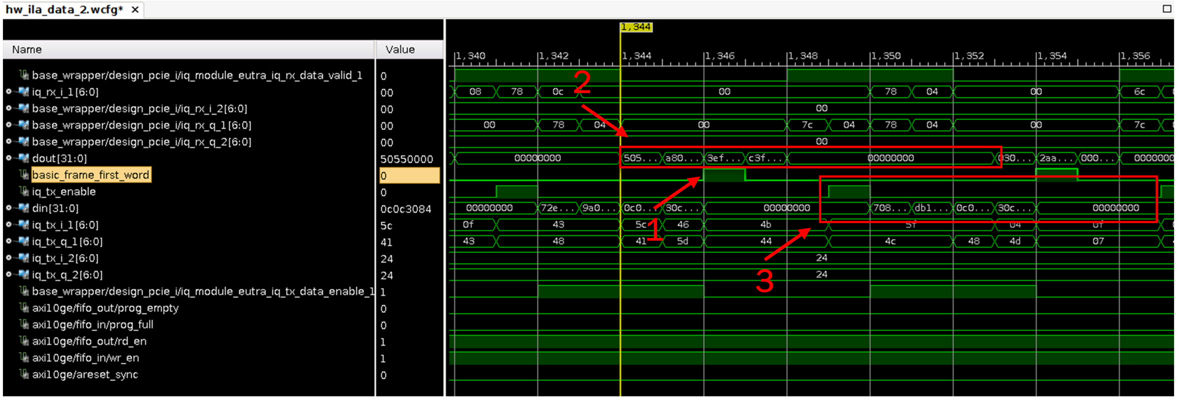


Figure 7.2: RF Spectrum transmitted by the AD9361.

The AD9361 was indeed outputting what seems to the naked eye the correct spectrum centered in 2.66 GHz (LTE band 7) and 10 MHz bandwidth as configured (Figure 7.2).

7.2.1.1 FIFO Overflow

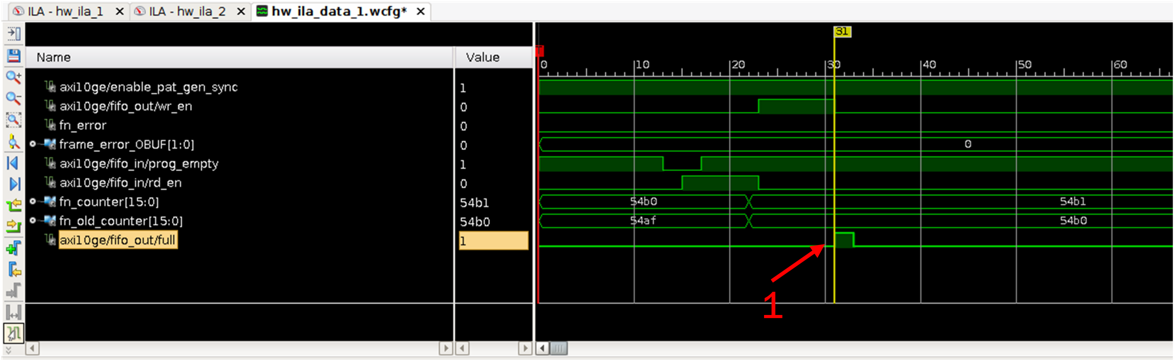
After careful analysis of captured data from the ILA debug cores it was discovered that there were periods of time when the *basic_frame_first_word* flag would desynchronize relatively to the received new basic frame (Figure 7.3). This phenomenon however, only manifested on the BBU VC707 while the RRH KC705 maintained a correct operation. Furthermore, the *iq_tx_enable* flag remained stable, indicating that the problem was specific to the uplink, either in RRH KC705 transmission, or BBU VC707 reception.



Name	Value
base_wrapper/design_pcie_i/iq_module_eutra_iq_rx_data_valid_1	0
iq_rx_i_1[6:0]	00
base_wrapper/design_pcie_i/iq_rx_i_2[6:0]	00
base_wrapper/design_pcie_i/iq_rx_q_1[6:0]	00
base_wrapper/design_pcie_i/iq_rx_q_2[6:0]	00
dout[31:0]	50550000
basic_frame_first_word	0
iq_tx_enable	0
din[31:0]	0c0c3084
iq_tx_i_1[6:0]	5c
iq_tx_q_1[6:0]	41
iq_tx_i_2[6:0]	24
iq_tx_q_2[6:0]	24
base_wrapper/design_pcie_i/iq_module_eutra_iq_tx_data_enable_1	1
axi10ge/fifo_out/prog_empty	0
axi10ge/fifo_in/prog_full	0
axi10ge/fifo_out/rd_en	1
axi10ge/fifo_in/wr_en	1
axi10ge/areset_sync	0

Figure 7.3: Loss of Synchronism at the BBU between *basic_frame_first_word* and new basic frame.

After further investigation on the ILA debug cores it was detected that the RX FIFO on the BBU side was overflowing and causing the synchronization problems. The RX FIFO full flag can be seen asserted in Figure 7.4.

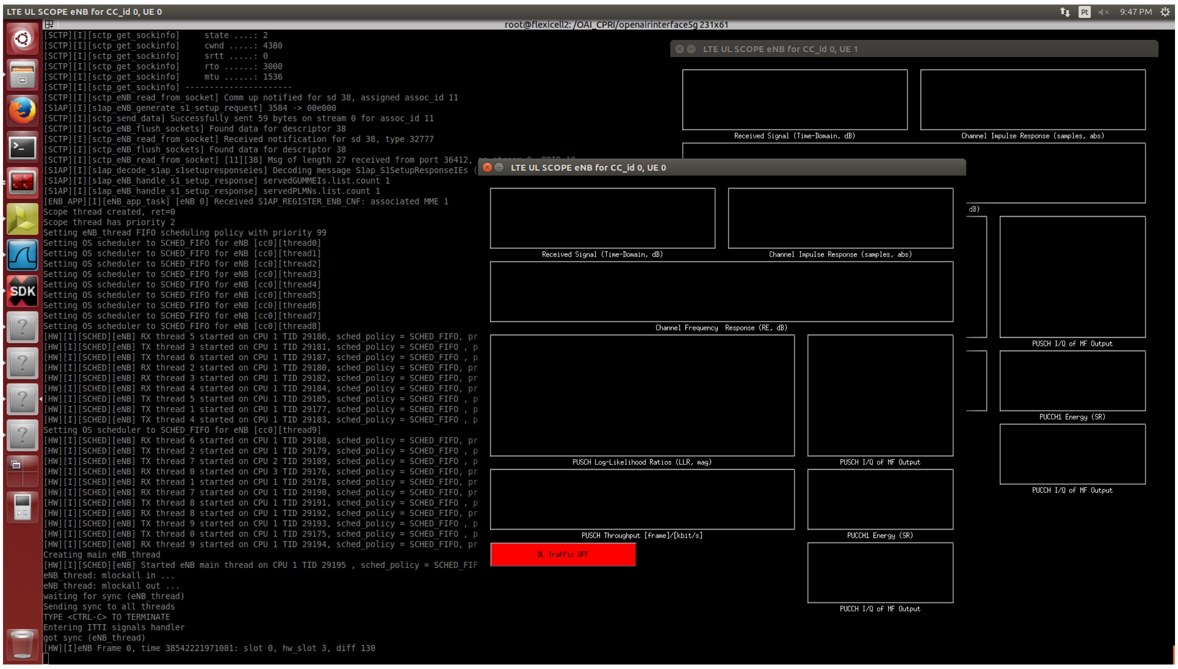


Name	Value
axi10ge/enable_pat_gen_sync	1
axi10ge/fifo_out/wr_en	0
fn_error	0
frame_error_OBUF[1:0]	0
axi10ge/fifo_in/prog_empty	1
axi10ge/fifo_in/rd_en	0
fn_counter[15:0]	54b1
fn_old_counter[15:0]	54b0
axi10ge/fifo_out/full	1

Figure 7.4: FIFO overflow on the BBU VC707 displayed by the ILA.

Since the 30.72 MHz EUTRA clock is being generated locally in each board, any slight variation on the nominal frequency will sooner or later translate in the overflow of one the FIFOs. To rectify this problem, a common 30.72 MHz clock is now used for both boards. The RRH design is modified to forward its locally generated 30.72 MHz clock from the FPGA logic to the output SMA connectors. This was done using the OBUFDS and ODDR primitives as recommended by Xilinx [Xil18a]. The use of the ODDR provides the best possible path and guarantees that the clock signal does not leave the global clock resources of the FPGA. The OBUFDS primitive is a differential output buffer that isolates the internal circuit and provides drive current for signals leaving the FPGA [Xil13]. The 30.72 MHz clock is then routed with a pair of SMA cables to the VC707 dedicated clock input SMA connectors. In the VC707 a IBUFDS primitive converts the differential input clock to single-ended and routes it to the FPGA logic.

Now with a common clock the RX FIFO do not overflow and correct operation was achieved on the BBU as well. When launching the OAI LTE soft-modem the RRH and BBU synchronized successfully. Figure 7.5 shows a screenshot of the OAI LTE soft-modem running with BBU and RRH synced and waiting for a UE to connect.

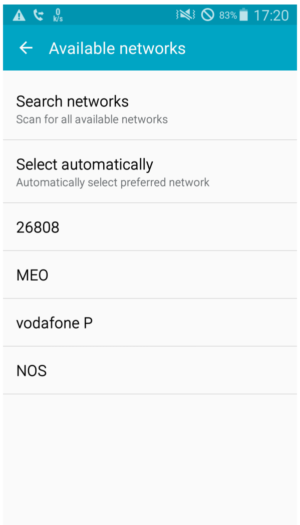

LTE UL SCOPE eNB for CC_id 0, UE 0
9:47 PM
root@flexicell2: /OAI_CPRI/openairinterface5g 231x61
LTE UL SCOPE eNB for CC_id 0, UE 1
Received Signal (Time-Domain, dB)
Channel Impulse Response (samples, abs)
LTE UL SCOPE eNB for CC_id 0, UE 0
Received Signal (Time-Domain, dB)
Channel Impulse Response (samples, abs)
Channel Frequency Response (RE, dB)
PUSCH Log-Likelihood Ratios (LLR, mag)
PUSCH I/Q of MF Output
PUSCH Throughput [frame]/[kbit/s]
PUCCH1 Energy (SR)
DL Traffic OFF
PUCCH I/Q of MF Output
dB)
PUSCH I/Q of MF Output
PUCCH1 Energy (SR)
PUCCH I/Q of MF Output

```
[SCTP][I][sctp_get_sockinfo]     state ....: 2
[SCTP][I][sctp_get_sockinfo]     cwnd .....: 4380
[SCTP][I][sctp_get_sockinfo]     srtt .....: 0
[SCTP][I][sctp_get_sockinfo]     rto ......: 3000
[SCTP][I][sctp_get_sockinfo]     mtu ......: 1536
[SCTP][I][sctp_get_sockinfo] ----------------------
[SCTP][I][sctp_eNB_read_from_socket] Comm up notified for sd 38, assigned assoc_id 11
[S1AP][I][s1ap_eNB_generate_s1_setup_request] 3584 -> 00e000
[SCTP][I][sctp_send_data] Successfully sent 59 bytes on stream 0 for assoc_id 11
[SCTP][I][sctp_eNB_flush_sockets] Found data for descriptor 38
[SCTP][I][sctp_eNB_read_from_socket] Received notification for sd 38, type 32777
[SCTP][I][sctp_eNB_flush_sockets] Found data for descriptor 38
[SCTP][I][sctp_eNB_read_from_socket] [11][38] Msg of length 27 received from port 36412,
[S1AP][I][s1ap_decode_s1ap_s1setupresponseies] Decoding message S1ap_S1SetupResponseIEs (
[S1AP][I][s1ap_eNB_handle_s1_setup_response] servedGUMMEIs.list.count 1
[S1AP][I][s1ap_eNB_handle_s1_setup_response] servedPLMNs.list.count 1
[ENB_APP][I][eNB_app_task] [eNB 0] Received S1AP_REGISTER_ENB_CNF: associated MME 1
Scope thread created, ret=0
Scope thread has priority 2
Setting eNB_thread FIFO scheduling policy with priority 99
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread0]
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread1]
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread2]
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread3]
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread4]
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread5]
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread6]
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread7]
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread8]
[HW][I][SCHED][eNB] RX thread 5 started on CPU 1 TID 29186, sched_policy = SCHED_FIFO, pr
[HW][I][SCHED][eNB] TX thread 3 started on CPU 1 TID 29181, sched_policy = SCHED_FIFO , p
[HW][I][SCHED][eNB] TX thread 6 started on CPU 1 TID 29187, sched_policy = SCHED_FIFO , p
[HW][I][SCHED][eNB] RX thread 2 started on CPU 1 TID 29180, sched_policy = SCHED_FIFO, pr
[HW][I][SCHED][eNB] RX thread 3 started on CPU 1 TID 29182, sched_policy = SCHED_FIFO, pr
[HW][I][SCHED][eNB] RX thread 4 started on CPU 1 TID 29184, sched_policy = SCHED_FIFO, pr
[HW][I][SCHED][eNB] TX thread 5 started on CPU 1 TID 29185, sched_policy = SCHED_FIFO , p
[HW][I][SCHED][eNB] TX thread 1 started on CPU 1 TID 29177, sched_policy = SCHED_FIFO , p
[HW][I][SCHED][eNB] TX thread 4 started on CPU 1 TID 29183, sched_policy = SCHED_FIFO , p
Setting OS scheduler to SCHED_FIFO for eNB [cc0][thread9]
[HW][I][SCHED][eNB] RX thread 6 started on CPU 1 TID 29188, sched_policy = SCHED_FIFO, pr
[HW][I][SCHED][eNB] TX thread 2 started on CPU 1 TID 29179, sched_policy = SCHED_FIFO , p
[HW][I][SCHED][eNB] TX thread 7 started on CPU 2 TID 29189, sched_policy = SCHED_FIFO , p
[HW][I][SCHED][eNB] RX thread 0 started on CPU 3 TID 29176, sched_policy = SCHED_FIFO, pr
[HW][I][SCHED][eNB] RX thread 1 started on CPU 1 TID 29178, sched_policy = SCHED_FIFO, pr
[HW][I][SCHED][eNB] RX thread 7 started on CPU 1 TID 29190, sched_policy = SCHED_FIFO, pr
[HW][I][SCHED][eNB] TX thread 8 started on CPU 1 TID 29191, sched_policy = SCHED_FIFO , p
[HW][I][SCHED][eNB] RX thread 8 started on CPU 1 TID 29192, sched_policy = SCHED_FIFO, pr
[HW][I][SCHED][eNB] TX thread 9 started on CPU 1 TID 29193, sched_policy = SCHED_FIFO , p
[HW][I][SCHED][eNB] TX thread 0 started on CPU 1 TID 29175, sched_policy = SCHED_FIFO , p
[HW][I][SCHED][eNB] RX thread 9 started on CPU 1 TID 29194, sched_policy = SCHED_FIFO, pr
Creating main eNB_thread
[HW][I][SCHED][eNB] Started eNB main thread on CPU 1 TID 29195 , sched_policy = SCHED_FIF
eNB_thread: mlockall in ...
eNB_thread: mlockall out ...
waiting for sync (eNB_thread)
Sending sync to all threads
TYPE <CTRL-C> TO TERMINATE
Entering ITTI signals handler
got sync (eNB_thread)
[HW][I]eNB Frame 0, time 38542221971081: slot 0, hw_slot 3, diff 130
```

Figure 7.5: OAI LTE soft-modem running with synced BBU/RRH

7.2.1.2 Reference Clocks

The BBU and RRH are now synced, and the smartphone could detect the network by performing a mobile network search. Figure 7.6 shows a screenshot of the smartphone after performing a mobile network search and finding the C-RAN along with commercial MNOs. The smartphone detects the mobile network as 26808, this are the Mobile Country Code (MCC) (268 - Portugal MCC) and Mobile Network Code (MNC) (08) concatenated.



Available networks

Search networks
Scan for all available networks

Select automatically
Automatically select preferred network

26808

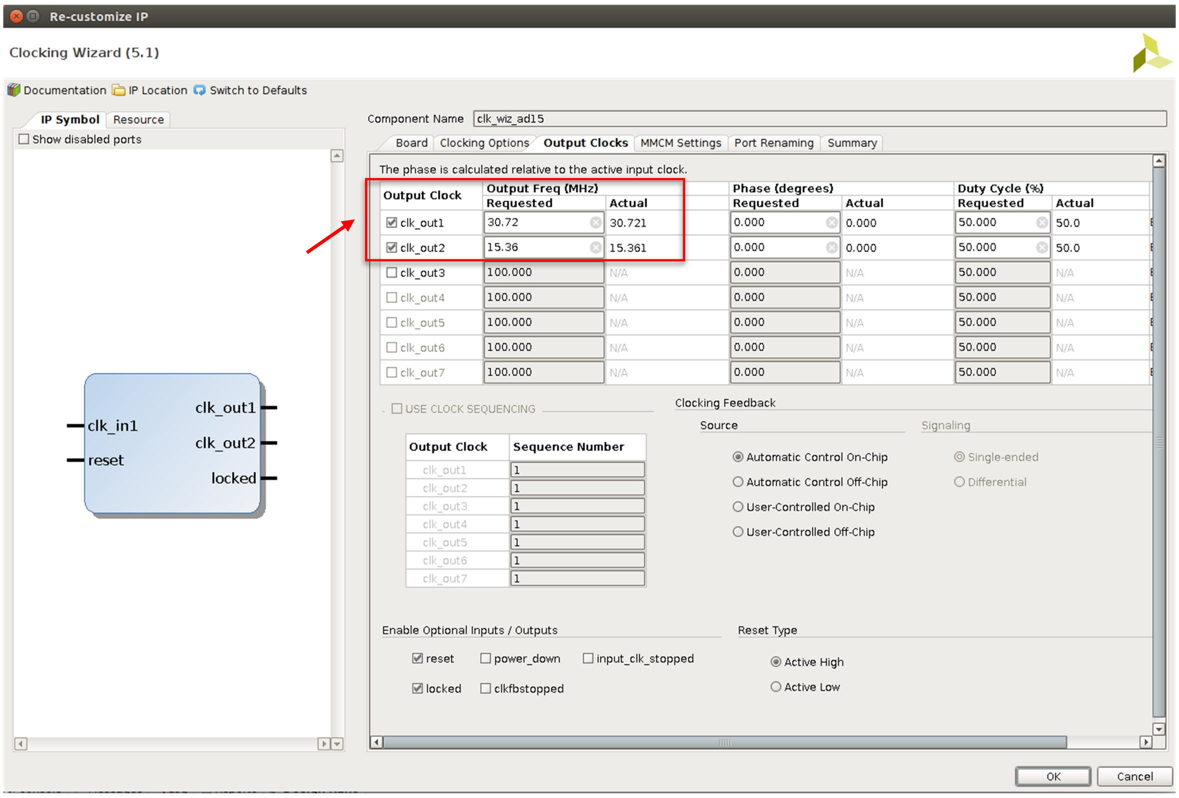
MEO

vodafone P

NOS

Figure 7.6: Smartphone detecting the mobile network.

Although the UE could now find the network, it would not connect to the BBU. After further investigation it was discovered that the KC705 MMCM was generating the wrong frequency. At this stage, both 30.72 MHz and 15.36 MHz clocks were generated in the RRH KC705 board using the Vivado Clocking Wizard IP core that implemented a single MMCM to generate both clocks. For some unknown reason, when both clocks were generated through the same MMCM, the actual output frequencies did not match the requested frequencies. This can be seen when parameterizing the IP core in Figure 7.7.



Clocking Wizard (5.1)

□ Show disabled ports

Component Name: clk_wiz_ad15

The phase is calculated relative to the active input clock.

Output Clock	Output Freq (MHz)		Phase (degrees)		Duty Cycle (%)	
	Requested	Actual	Requested	Actual	Requested	Actual
☑ clk_out1	30.72	30.721	0.000	0.000	50.000	50.0
☑ clk_out2	15.36	15.361	0.000	0.000	50.000	50.0
□ clk_out3	100.000	N/A	0.000	N/A	50.000	N/A
□ clk_out4	100.000	N/A	0.000	N/A	50.000	N/A
□ clk_out5	100.000	N/A	0.000	N/A	50.000	N/A
□ clk_out6	100.000	N/A	0.000	N/A	50.000	N/A
□ clk_out7	100.000	N/A	0.000	N/A	50.000	N/A

□ USE CLOCK SEQUENCING

Output Clock	Sequence Number
clk_out1	1
clk_out2	1
clk_out3	1
clk_out4	1
clk_out5	1
clk_out6	1
clk_out7	1

Clocking Feedback

Source
● Automatic Control On-Chip
○ Automatic Control Off-Chip
○ User-Controlled On-Chip
○ User-Controlled Off-Chip

Signaling
● Single-ended
○ Differential

Enable Optional Inputs / Outputs
☑ reset □ power_down □ input_clk_stopped
☑ locked □ clkfbstopped

Reset Type
● Active High
○ Active Low

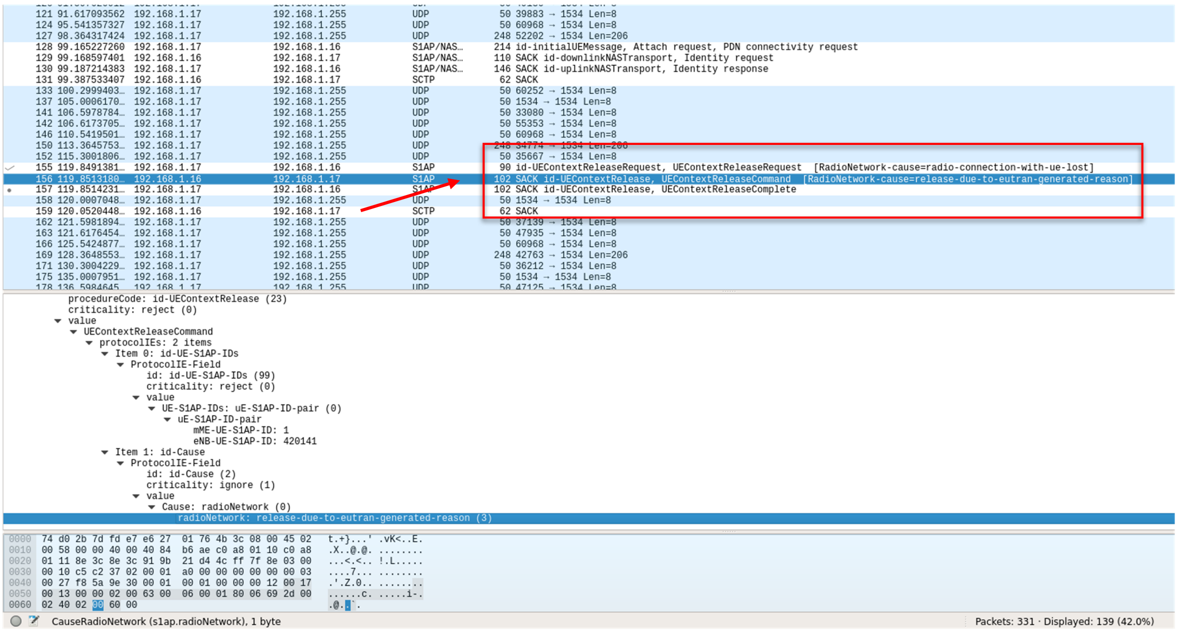
Figure 7.7: Vivado Clocking Wizard IP Configuration.

After implementing distinct MMCMs for each output clock, the actual output frequencies were corrected and the smartphone would manage to connect to the BBU and attach briefly to the EPC before crashing.

7.2.1.3 AD9361 RF Transceiver Reference Clock

Analysis of OAI LTE soft-modem output logs and Wireshark traces captured in the S1 interface (between EPC and BBU) indicate that shortly after the smartphone attach to the EPC, the EPC sends a command for the RRH to release the UE, indicating that radio connection with the UE was lost (Figure 7.8). This means that there is still some problem related to the RF front-end in the RRH.

Although not quantified, since previous problems were due to reference clocks, it was tried a new approach regarding the 30.72 MHz EUTRA clocks and 15.36 MHz reference clock for the AD9361.



```
121 91.617093562 192.168.1.17      192.168.1.255     UDP           50 39883 → 1534 Len=8
124 95.541357327 192.168.1.17      192.168.1.255     UDP           50 60968 → 1534 Len=8
127 98.364317424 192.168.1.17      192.168.1.255     UDP          248 52202 → 1534 Len=206
128 99.165227260 192.168.1.17      192.168.1.16      S1AP/NAS…    214 id-initialUEMessage, Attach request, PDN connectivity request
129 99.168597401 192.168.1.16      192.168.1.17      S1AP/NAS…    110 SACK id-downlinkNASTransport, Identity request
130 99.187214383 192.168.1.17      192.168.1.16      S1AP/NAS…    146 SACK id-uplinkNASTransport, Identity response
131 99.387533407 192.168.1.16      192.168.1.17      SCTP          62 SACK
133 100.2999403… 192.168.1.17      192.168.1.255     UDP           50 60252 → 1534 Len=8
137 105.0006170… 192.168.1.17      192.168.1.255     UDP           50 1534 → 1534 Len=8
141 106.5978784… 192.168.1.17      192.168.1.255     UDP           50 33080 → 1534 Len=8
142 106.6173705… 192.168.1.17      192.168.1.255     UDP           50 55353 → 1534 Len=8
146 110.5419501… 192.168.1.17      192.168.1.255     UDP           50 60968 → 1534 Len=8
150 113.3645753… 192.168.1.17      192.168.1.255     UDP          248 34774 → 1534 Len=206
152 115.3001806… 192.168.1.17      192.168.1.255     UDP           50 35667 → 1534 Len=8
155 119.8491381… 192.168.1.17      192.168.1.16      S1AP          90 id-UEContextReleaseRequest, UEContextReleaseRequest  [RadioNetwork-cause=radio-connection-with-ue-lost]
156 119.8513180… 192.168.1.16      192.168.1.17      S1AP         102 SACK id-UEContextRelease, UEContextReleaseCommand  [RadioNetwork-cause=release-due-to-eutran-generated-reason]
157 119.8514231… 192.168.1.17      192.168.1.16      S1AP         102 SACK id-UEContextRelease, UEContextReleaseComplete
158 120.0007048… 192.168.1.17      192.168.1.255     UDP           50 1534 → 1534 Len=8
159 120.0520448… 192.168.1.16      192.168.1.17      SCTP          62 SACK
162 121.5981894… 192.168.1.17      192.168.1.255     UDP           50 37139 → 1534 Len=8
163 121.6176454… 192.168.1.17      192.168.1.255     UDP           50 47935 → 1534 Len=8
166 125.5424877… 192.168.1.17      192.168.1.255     UDP           50 60968 → 1534 Len=8
169 128.3648553… 192.168.1.17      192.168.1.255     UDP          248 42763 → 1534 Len=206
171 130.3004229… 192.168.1.17      192.168.1.255     UDP           50 36212 → 1534 Len=8
175 135.0007951… 192.168.1.17      192.168.1.255     UDP           50 1534 → 1534 Len=8
178 136.5984645  192.168.1.17      192.168.1.255     UDP           50 47125 → 1534 Len=8
```

```
      procedureCode: id-UEContextRelease (23)
      criticality: reject (0)
    ▾ value
      ▾ UEContextReleaseCommand
        ▾ protocolIEs: 2 items
          ▾ Item 0: id-UE-S1AP-IDs
            ▾ ProtocolIE-Field
                id: id-UE-S1AP-IDs (99)
                criticality: reject (0)
              ▾ value
                ▾ UE-S1AP-IDs: uE-S1AP-ID-pair (0)
                  ▾ uE-S1AP-ID-pair
                      mME-UE-S1AP-ID: 1
                      eNB-UE-S1AP-ID: 420141
          ▾ Item 1: id-Cause
            ▾ ProtocolIE-Field
                id: id-Cause (2)
                criticality: ignore (1)
              ▾ value
                ▾ Cause: radioNetwork (0)
                    radioNetwork: release-due-to-eutran-generated-reason (3)
```

```
0000  74 d0 2b 7d fd e7 e6 27  01 76 4b 3c 08 00 45 02   t.+}...' .vK<..E.
0010  00 58 00 00 40 00 40 84  b6 ae c0 a8 01 10 c0 a8   .X..@.@. ........
0020  01 11 8e 3c 8e 3c 91 9b  21 d4 4c ff 7f 8e 03 00   ...<.<.. !.L.....
0030  00 10 c5 c2 37 02 00 01  a0 00 00 00 00 00 00 03   ....7... ........
0040  00 27 f8 5a 9e 30 00 01  00 01 00 00 00 12 00 17   .'.Z.0.. ........
0050  00 13 00 00 02 00 63 00  06 00 01 80 06 69 2d 00   ......c. .....i-.
0060  02 40 02 00 60 00                                  .@..`.
```

CauseRadioNetwork (s1ap.radioNetwork), 1 byte   Packets: 331 · Displayed: 139 (42.0%)

Figure 7.8: Wireshark traces and LTE soft-modem output log.

Instead of generating the aforementioned clocks within the KC705 board, a single reference clock of 15.36 MHz is generated in a external signal generator (Rohde&Schwarz SMC100A). This more precise reference clock is then used as an input to the CDCE72010EVM low jitter clock synchronizer and used to generate the 30.72 MHz clock for both KC705 and VC707 boards as well as the 61.44 MHz reference clock for the AD9361 RF transceiver.

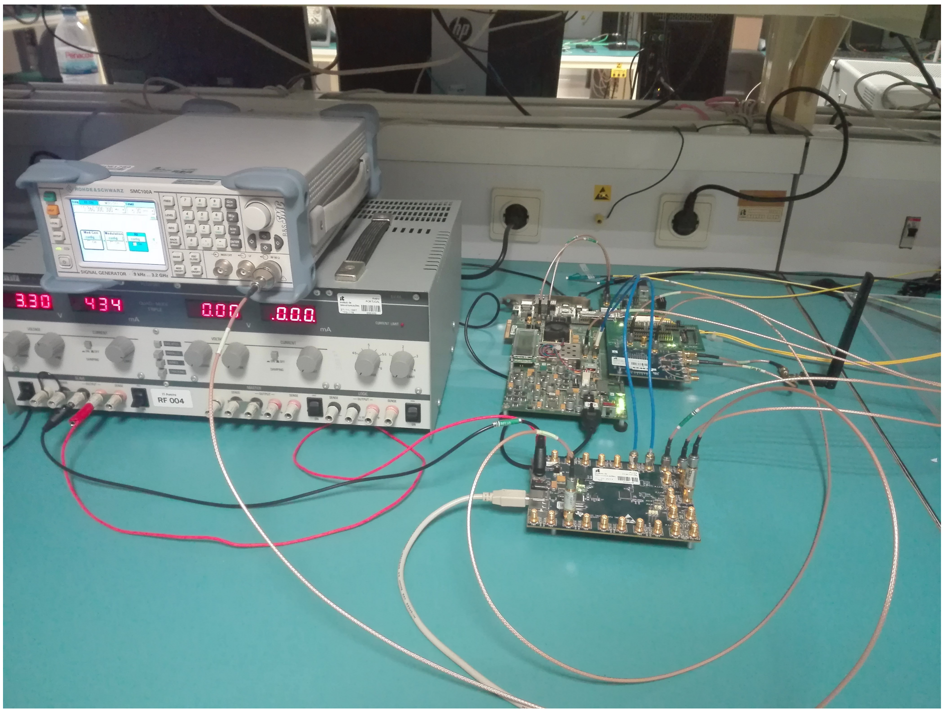


Figure 7.9: Picture of the RRH setup.

The new RRH setup is depicted in Figure 7.9. In the figure is shown the SMC100A signal generator generating the 15.36 MHz input clock to the CDCE72010EVM. The low jitter clock synchronizer then generates two pairs of LVDS 30.72 MHz clock signals for the KC705 and VC707 (top of the CDCE72010EVM) and a single LVCMOS 61.44 MHz reference clock signal for the AD9361 RF transceiver. Figure 7.10 shows the graphical user interface used to program the CDCE72010EVM module with new parameters.

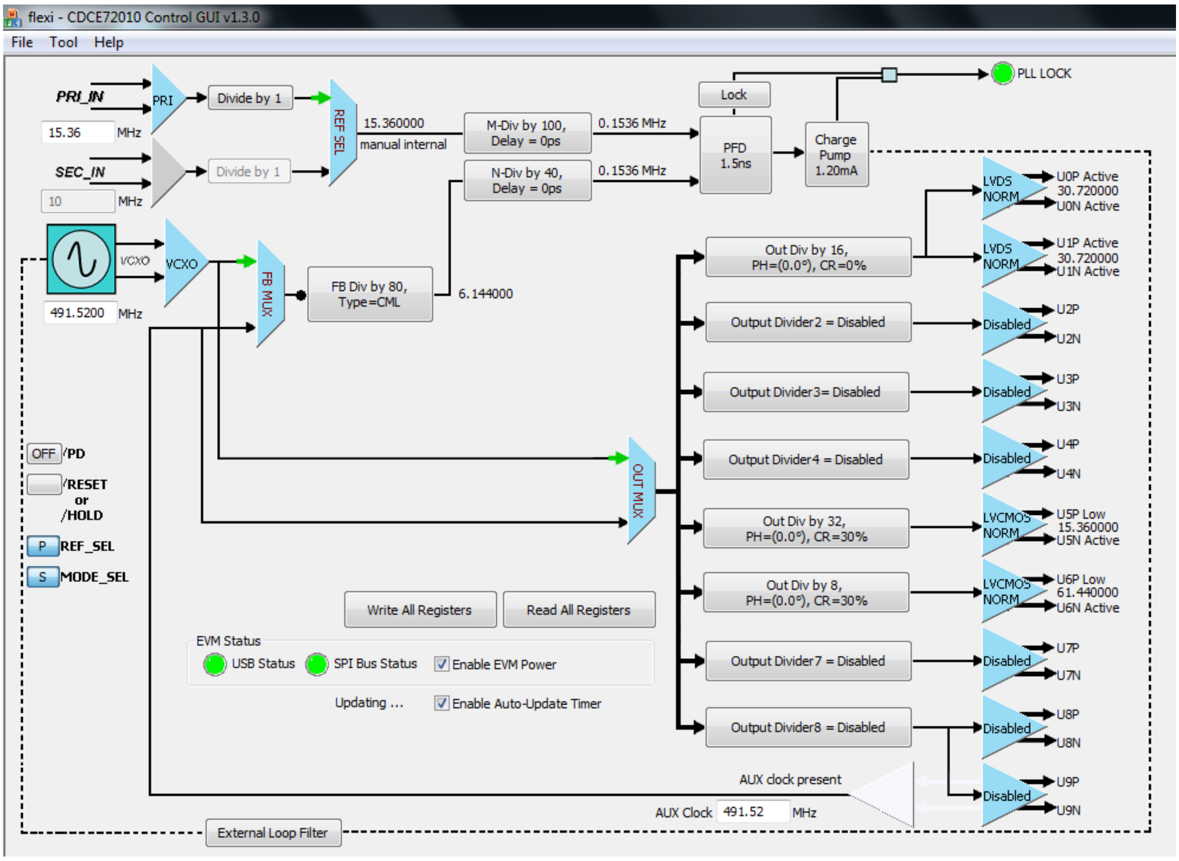
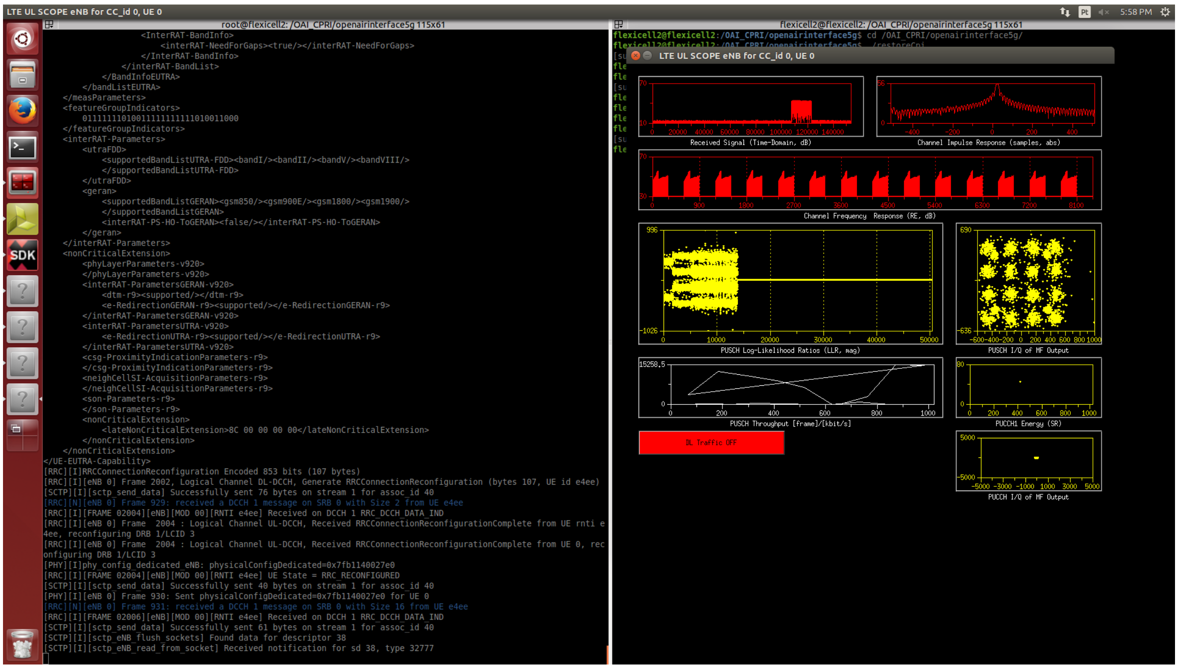


Figure 7.10: CDCE72010EVM GUI.

With these modifications the smartphone can finally connect and attach successfully to the C-RAN. The main objective of this dissertation was achieved. Figure 7.11 illustrates the OAI LTE soft-modem output (on the left) indicating that the UE is successfully connected and uplink scope (on the right) showing the constellation diagram, channel frequency response and other LTE parameters.

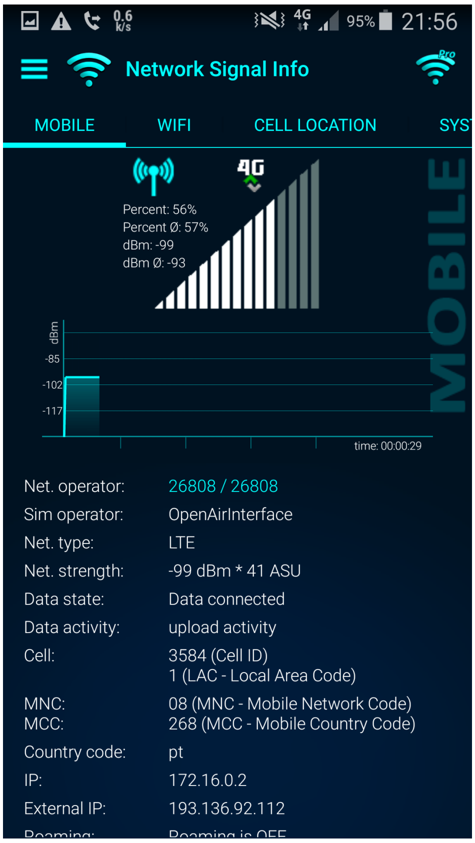
Figure 7.12 shows the network signal information of the smartphone successfully connected.


LTE UL SCOPE eNB for CC_id 0, UE 0
5:58 PM
root@flexicell2: /OAI_CPRI/openairinterface5g 115x61
flexicell2@flexicell2: /OAI_CPRI/openairinterface5g 115x61
LTE UL SCOPE eNB for CC_id 0, UE 0
70
10
0
20000
40000
60000
80000
100000
120000
140000
Received Signal (Time-Domain, dB)
56
0
-400
-200
0
200
400
Channel Impulse Response (samples, abs)
70
30
0
900
1800
2700
3600
4500
5400
6300
7200
8100
Channel Frequency Response (RE, dB)
996
-1026
0
10000
20000
30000
40000
50000
PUSCH Log-Likelihood Ratios (LLR, mag)
690
-636
-600-400-200 0 200 400 600 800 1000
PUSCH I/Q of MF Output
15258.5
0
0
200
400
600
800
1000
PUSCH Throughput [frame]/[kbit/s]
80
0
0
200
400
600
800
1000
PUCCH1 Energy (SR)
DL Traffic OFF
5000
-5000
-5000 -3000 -1000 1000 3000 5000
PUCCH I/Q of MF Output

```
                <InterRAT-BandInfo>
                    <interRAT-NeedForGaps><true/></interRAT-NeedForGaps>
                </InterRAT-BandInfo>
            </interRAT-BandList>
        </BandInfoEUTRA>
    </bandListEUTRA>
</measParameters>
<featureGroupIndicators>
    01111111010011111111111010011000
</featureGroupIndicators>
<interRAT-Parameters>
    <utraFDD>
        <supportedBandListUTRA-FDD><bandI/><bandII/><bandV/><bandVIII/>
        </supportedBandListUTRA-FDD>
    </utraFDD>
    <geran>
        <supportedBandListGERAN><gsm850/><gsm900E/><gsm1800/><gsm1900/>
        </supportedBandListGERAN>
        <interRAT-PS-HO-ToGERAN><false/></interRAT-PS-HO-ToGERAN>
    </geran>
</interRAT-Parameters>
<nonCriticalExtension>
    <phyLayerParameters-v920>
    </phyLayerParameters-v920>
    <interRAT-ParametersGERAN-v920>
        <dtm-r9><supported/></dtm-r9>
        <e-RedirectionGERAN-r9><supported/></e-RedirectionGERAN-r9>
    </interRAT-ParametersGERAN-v920>
    <interRAT-ParametersUTRA-v920>
        <e-RedirectionUTRA-r9><supported/></e-RedirectionUTRA-r9>
    </interRAT-ParametersUTRA-v920>
    <csg-ProximityIndicationParameters-r9>
    </csg-ProximityIndicationParameters-r9>
    <neighCellSI-AcquisitionParameters-r9>
    </neighCellSI-AcquisitionParameters-r9>
    <son-Parameters-r9>
    </son-Parameters-r9>
    <nonCriticalExtension>
        <lateNonCriticalExtension>8C 00 00 00 00</lateNonCriticalExtension>
    </nonCriticalExtension>
</nonCriticalExtension>
</UE-EUTRA-Capability>
[RRC][I]RRCConnectionReconfiguration Encoded 853 bits (107 bytes)
[RRC][I][eNB 0] Frame 2002, Logical Channel DL-DCCH, Generate RRCConnectionReconfiguration (bytes 107, UE id e4ee)
[SCTP][I][sctp_send_data] Successfully sent 76 bytes on stream 1 for assoc_id 40
[RRC][N][eNB 0] Frame 929: received a DCCH 1 message on SRB 0 with Size 2 from UE e4ee
[RRC][I][FRAME 02004][eNB][MOD 00][RNTI e4ee] Received on DCCH 1 RRC_DCCH_DATA_IND
[RRC][I][eNB 0] Frame  2004 : Logical Channel UL-DCCH, Received RRCConnectionReconfigurationComplete from UE rnti e
4ee, reconfiguring DRB 1/LCID 3
[RRC][I][eNB 0] Frame  2004 : Logical Channel UL-DCCH, Received RRCConnectionReconfigurationComplete from UE 0, rec
onfiguring DRB 1/LCID 3
[PHY][I]phy_config_dedicated_eNB: physicalConfigDedicated=0x7fb1140027e0
[RRC][I][FRAME 02004][eNB][MOD 00][RNTI e4ee] UE State = RRC_RECONFIGURED
[SCTP][I][sctp_send_data] Successfully sent 40 bytes on stream 1 for assoc_id 40
[PHY][I][eNB 0] Frame 930: Sent physicalConfigDedicated=0x7fb1140027e0 for UE 0
[RRC][N][eNB 0] Frame 931: received a DCCH 1 message on SRB 0 with Size 16 from UE e4ee
[RRC][I][FRAME 02006][eNB][MOD 00][RNTI e4ee] Received on DCCH 1 RRC_DCCH_DATA_IND
[SCTP][I][sctp_send_data] Successfully sent 61 bytes on stream 1 for assoc_id 40
[SCTP][I][sctp_eNB_flush_sockets] Found data for descriptor 38
[SCTP][I][sctp_eNB_read_from_socket] Received notification for sd 38, type 32777
```

```
flexicell2@flexicell2:/OAI_CPRI/openairinterface5g$ cd /OAI_CPRI/openairinterface5g/
flexicell2@flexicell2:/OAI_CPRI/openairinterface5g$ ./restoreCpi
```

Figure 7.11: Smartphone successfully attached to the C-RAN.



Percent: 56%
Percent Ø: 57%
dBm: -99
dBm Ø: -93

Net. operator: 26808 / 26808
Sim operator: OpenAirInterface
Net. type: LTE
Net. strength: -99 dBm * 41 ASU
Data state: Data connected
Data activity: upload activity
Cell: 3584 (Cell ID)
1 (LAC - Local Area Code)
MNC: 08 (MNC - Mobile Network Code)
MCC: 268 (MCC - Mobile Country Code)
Country code: pt
IP: 172.16.0.2
External IP: 193.136.92.112
Roaming: Roaming is OFF

Figure 7.12: Network signal information from the smartphone.

7.3 Latency Measurement and RX/TX Adjustment

In order for the UEs to connect and attach properly to the network, the TX and RX paths need to be synchronized at the BBU. This is done by adjusting a pointer in the embedded application running on top of the MicroBlaze soft-core processor. This pointer changes the position where incoming received frames are stored in a circular buffer before being consumed by the OAI LTE soft-modem through the PCIe interface, effectively compensating the latency between downlink and uplink paths.

In order to compensate the TX/RX latency it is necessary to determine its value. To measure the latency between RRH and BBU, additional logic was developed on top of the digital design. The procedure is the following:

1. While the RAN is running, a push-button on the RRH KC705 board is pressed to insert a known sequence (0x55) directly into EUTRA inputs.
2. At the same time that the sequence is inserted in the FIFOs, a flag is asserted an sent over a SMA cable directly from the RRH KC705 to the BBU VC707.
3. At the BBU side, the VC707 is already waiting for the flag signal as a trigger to capture data over the ILA debug core.
4. Latency is calculated by determining how many clock cycles have passed between the trigger event and the known sequence arrived at the output of the EUTRA module at the BBU.

Figure 7.13 tries to illustrate the previous procedure.

A screenshot of the ILA can be seen in Figure 7.14. Since the data capture begins when the flag signal is triggered, the latency value is equal to the number of clock cycles that took until the injected sequence of 0x55 is first seen at the output of the EUTRA module. In this case with the 20 km fronthaul was obtained 3032 clock cycles (ILA running at 30.72 MHz) from RRH to BBU. This translates to roughly 100 μs. The round trip delay is approximately 200 μs as the measured latency from BBU to RRH is roughly the same at 3036 clock cycles.

Knowing the latency value, the pointer is adjusted by 1 for each 66.667 μs (OFDM symbol duration). Finally a simple try and error strategy is used to fine tune the perfect value.

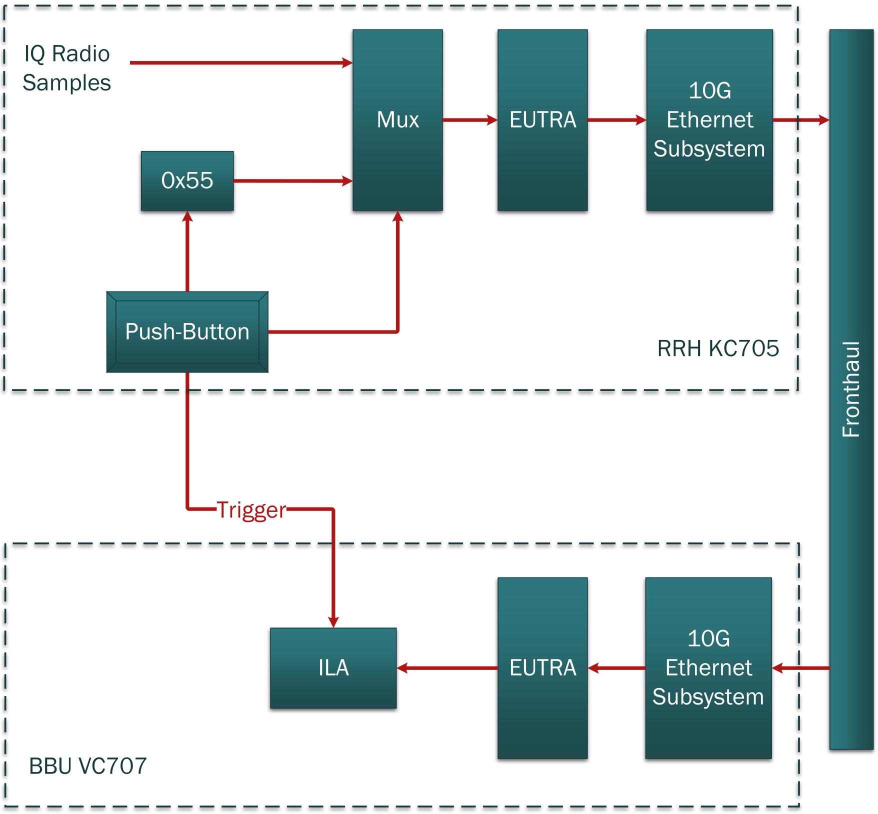
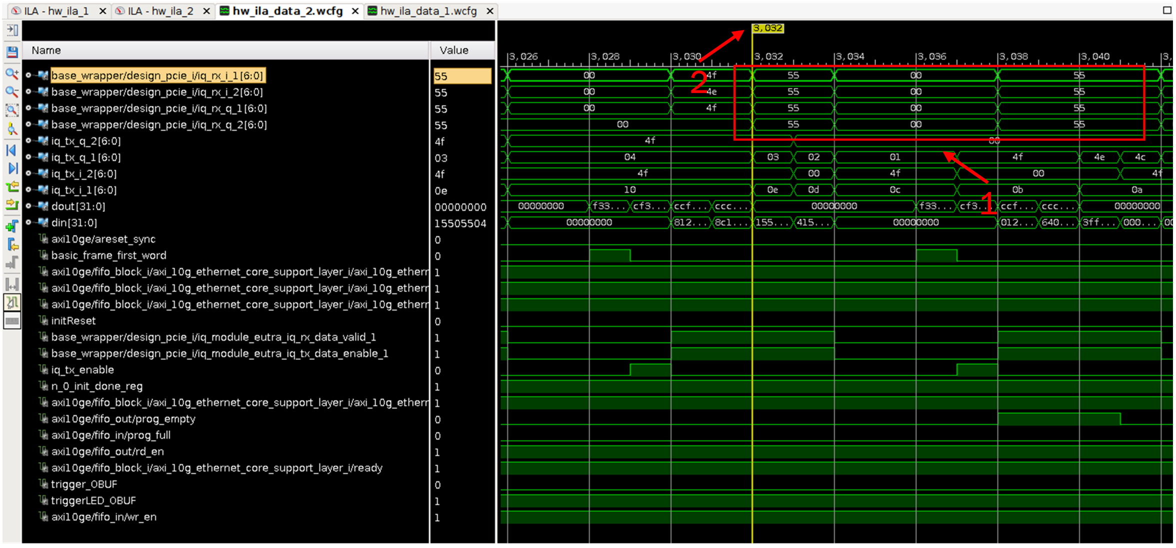


Figure 7.13: Block diagram of the logic involved in the latency measurement between RRH and BBU.



Name	Value
base_wrapper/design_pcie_i/iq_rx_i_1[6:0]	55
base_wrapper/design_pcie_i/iq_rx_i_2[6:0]	55
base_wrapper/design_pcie_i/iq_rx_q_1[6:0]	55
base_wrapper/design_pcie_i/iq_rx_q_2[6:0]	55
iq_tx_q_2[6:0]	4f
iq_tx_q_1[6:0]	03
iq_tx_i_2[6:0]	4f
iq_tx_i_1[6:0]	0e
dout[31:0]	00000000
din[31:0]	15505504
axi10ge/areset_sync	0
basic_frame_first_word	0
axi10ge/fifo_block_i/axi_10g_ethernet_core_support_layer_i/axi_10g_ethern	1
axi10ge/fifo_block_i/axi_10g_ethernet_core_support_layer_i/axi_10g_ethern	1
axi10ge/fifo_block_i/axi_10g_ethernet_core_support_layer_i/axi_10g_ethern	1
initReset	0
base_wrapper/design_pcie_i/iq_module_eutra_iq_rx_data_valid_1	1
base_wrapper/design_pcie_i/iq_module_eutra_iq_tx_data_enable_1	1
iq_tx_enable	0
n_0_init_done_reg	1
axi10ge/fifo_block_i/axi_10g_ethernet_core_support_layer_i/axi_10g_ethern	1
axi10ge/fifo_out/prog_empty	0
axi10ge/fifo_in/prog_full	0
axi10ge/fifo_out/rd_en	1
axi10ge/fifo_block_i/axi_10g_ethernet_core_support_layer_i/ready	1
trigger_OBUF	0
triggerLED_OBUF	1
axi10ge/fifo_in/wr_en	1

Figure 7.14: Latency measurement from RRH to BBU using the Integrated Logic Analyzer (ILA).

7.4 FPGA Resource Utilization

In this section are compared the FPGAs resource usage of the original CPRI-based digital design implementation with the 10GE developed one. Figure 7.15 shows the area occupied by each design in the RRH KC705 FPGA (Kintex-7) and Figure 7.16 illustrates graphically the amount of FPGA resources used in each implementation.

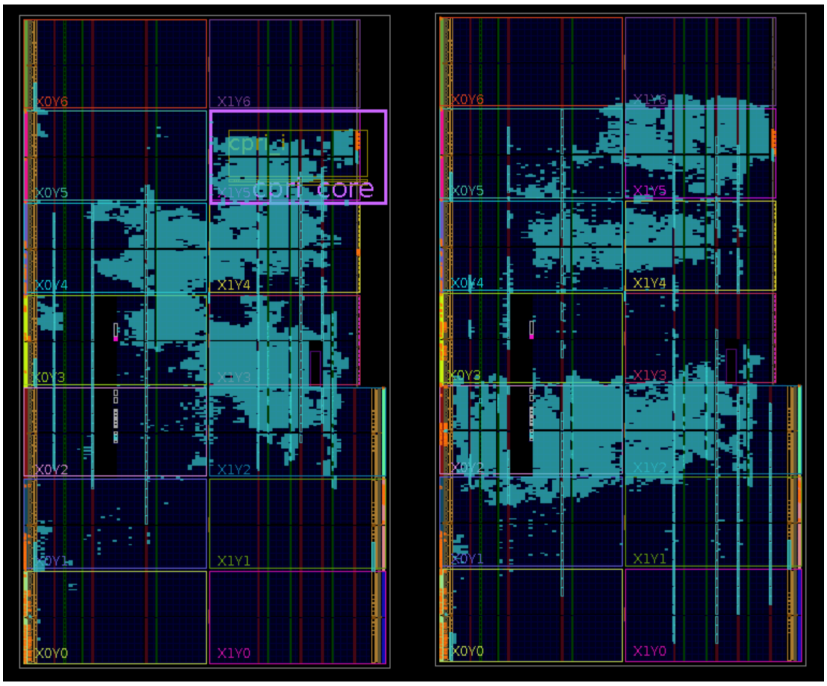


Figure 7.15: Kintex-7 FPGA implemented design - CPRI (left) vs 10GE (right).

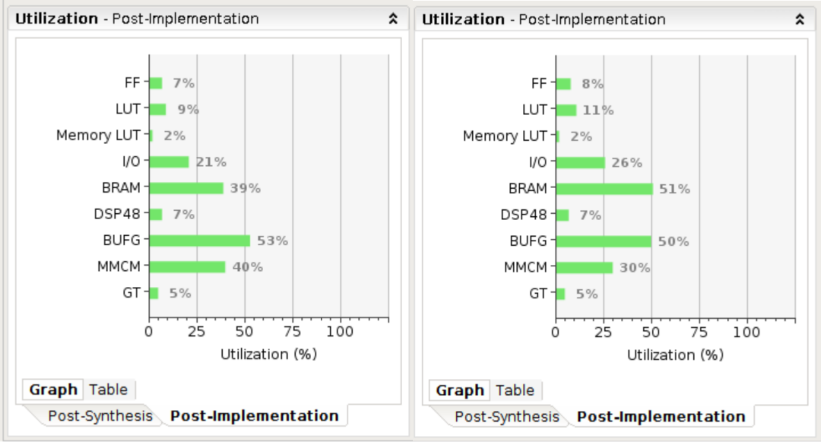


Figure 7.16: Kintex-7 FPGA resource usage comparison - CPRI (left) vs 10GE (right).

Figure 7.17 shows the area occupied by each design in the BBU VC707 FPGA (Virtex-7) and Figure 7.18 illustrates graphically the amount of FPGA resources used in each implementation.

As shown is previous figures, the FPGA resource utilization is very similar for both designs and as such the utilized area and respective power consumption will be identical as well. By

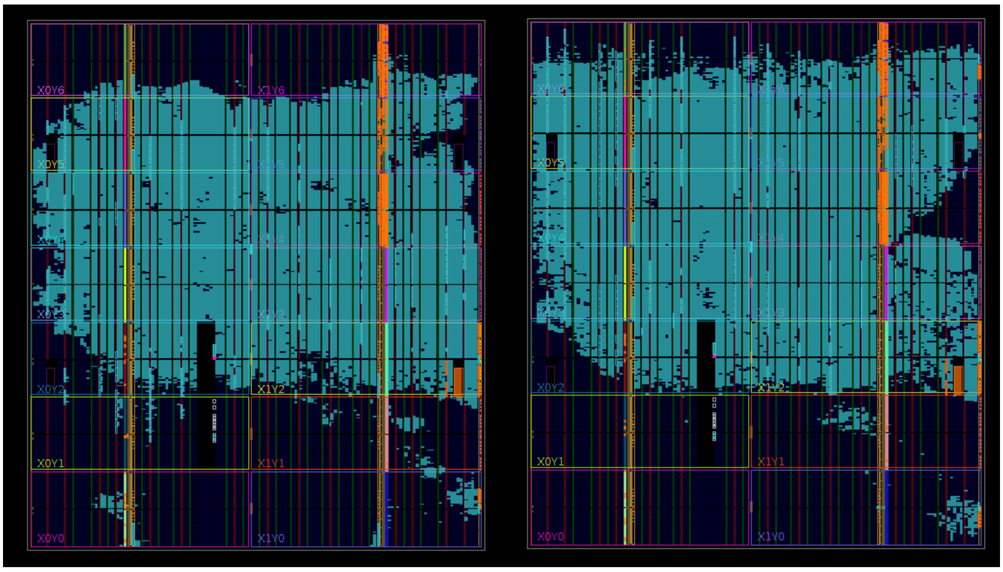


Figure 7.17: Virtex-7 FPGA implemented design - CPRI (left) vs 10GE (right).

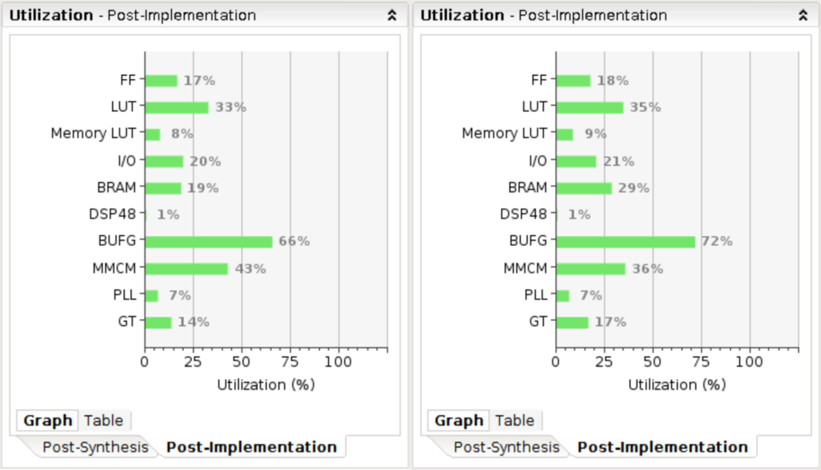


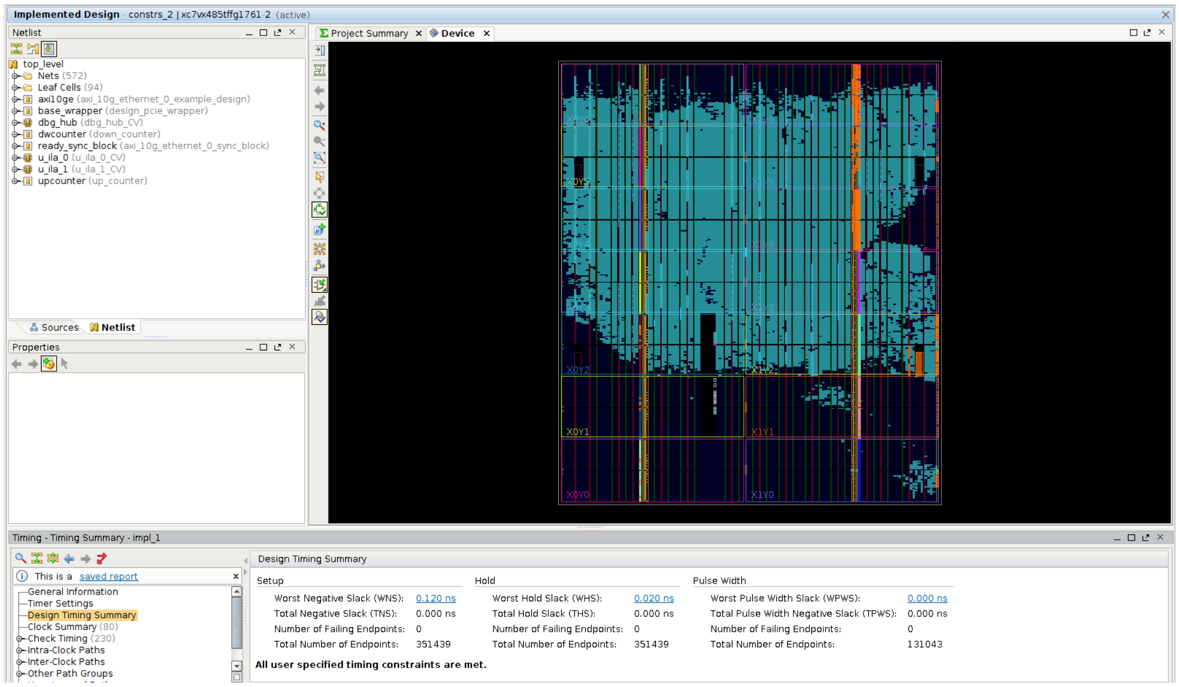
Figure 7.18: Virtex-7 FPGA resource usage comparison - CPRI (left) vs 10GE (right).

comparing Figure 7.15 with Figure 7.17 is displayed the difference between the Kintex-7 and Virtex-7 FPGA, with the Virtex-7 being much larger, packing more available resources. It can also be concluded that in both 10GE digital designs, the FPGAs still have some available resources left that could be used to perform a function split possibly including the lower LTE PHY layers such as FFT/IFFT blocks. This could greatly reduce the required fronthaul bandwidth and OpenAirInterface already supports this specific function split.

7.5 Timing Constraints

As mention briefly in the previous chapter, ILA debug cores were used extensively at this stage of the project to verify correct functionality and debugging support. However, the introduction of ILA cores led to some challenges primarily on the BBU digital design in the form of timing violations e.g., worst and total negative slack as a result of FPGA resources usage.

One way to circumvent this obstacle is to alter the implementation strategy used by Vivado. By default, Vivado utilizes an implementation strategy that tries to balance runtime with timing closure but the user can choose different strategies to improve performance, area and power [Xil14c]. Adopting a specific strategy does not guarantee however, definitive solutions i.e., different strategies can provide better results with distinctive designs. In this specific case it was found that the *Performance_Explore* and *Performance_ExplorePostRoutePhys* strategies delivered the best results.



Implemented Design - constrs_2 | xc7vx485tffg1761-2 (active)

Netlist

```
top_level
  Nets (572)
  Leaf Cells (94)
  axi10ge (axi_10g_ethernet_0_example_design)
  base_wrapper (design_pcie_wrapper)
  dbg_hub (dbg_hub_CV)
  dwcounter (down_counter)
  ready_sync_block (axi_10g_ethernet_0_sync_block)
  u_ila_0 (u_ila_0_CV)
  u_ila_1 (u_ila_1_CV)
  upcounter (up_counter)
```

Timing - Timing Summary - impl_1

This is a saved report

```
General Information
Timer Settings
Design Timing Summary
Clock Summary (80)
Check Timing (230)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
```

Design Timing Summary

```
Setup                                      Hold                                       Pulse Width
  Worst Negative Slack (WNS):   0.120 ns     Worst Hold Slack (WHS):       0.020 ns     Worst Pulse Width Slack (WPWS):             0.000 ns
  Total Negative Slack (TNS):   0.000 ns     Total Hold Slack (THS):       0.000 ns     Total Pulse Width Negative Slack (TPWS):    0.000 ns
  Number of Failing Endpoints:  0            Number of Failing Endpoints:  0            Number of Failing Endpoints:                0
  Total Number of Endpoints:    351439       Total Number of Endpoints:    351439       Total Number of Endpoints:                  131043
All user specified timing constraints are met.
```

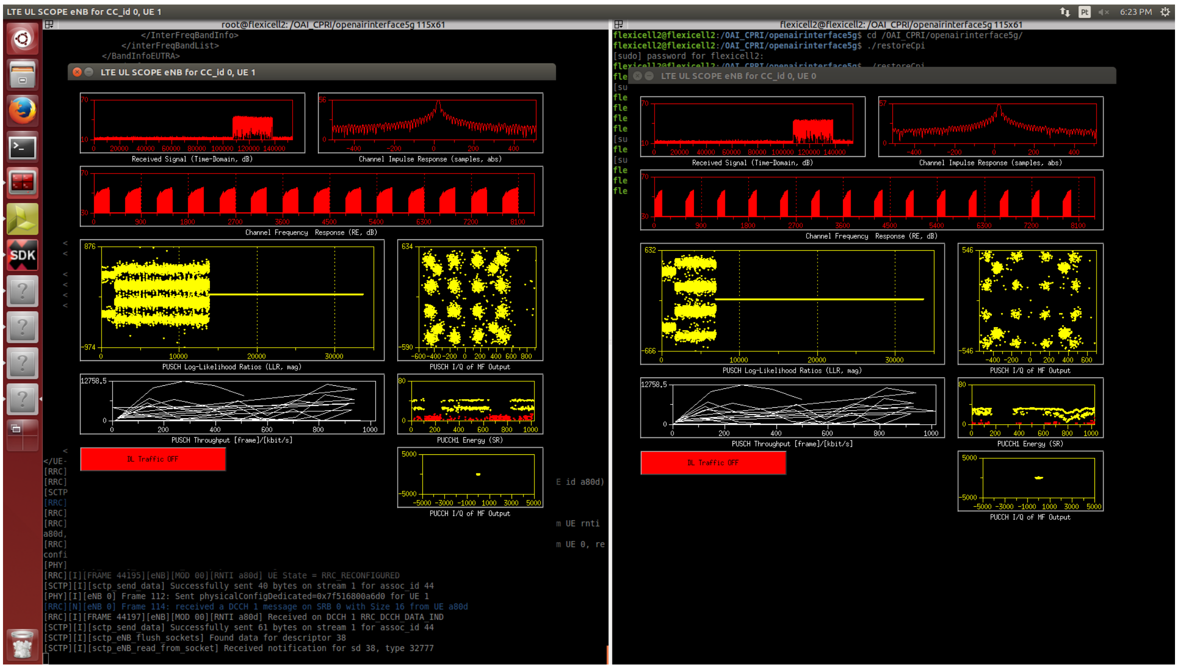
Figure 7.19: BBU implemented design.

Another solution is to place well defined constraints in the project XDC file, mainly in all clock signals to ensure that the synthesis and implementation tool on Vivado can appropriately place the required blocks and route the clocks in order to ensure that all timing requirements are met. On Figure 7.19 is shown that the Virtex-7 FPGA has already more than 70% of its area populated. On the bottom of the image can be seen that the implementation was achieved with all timing constraints met.

7.6 Performance Tests

By the end, with the integration of the 10G Ethernet fronthaul solution successfully merged with the original C-RAN test-bed a few performance tests were made. These tests should be considered merely informative of system stability since the new fronthaul interface only by itself won't bring any performance improvements when compared to the original CPRI setup as the RF equipment and configuration are still the same.

Figure 7.20 shows two user equipments connected simultaneously.

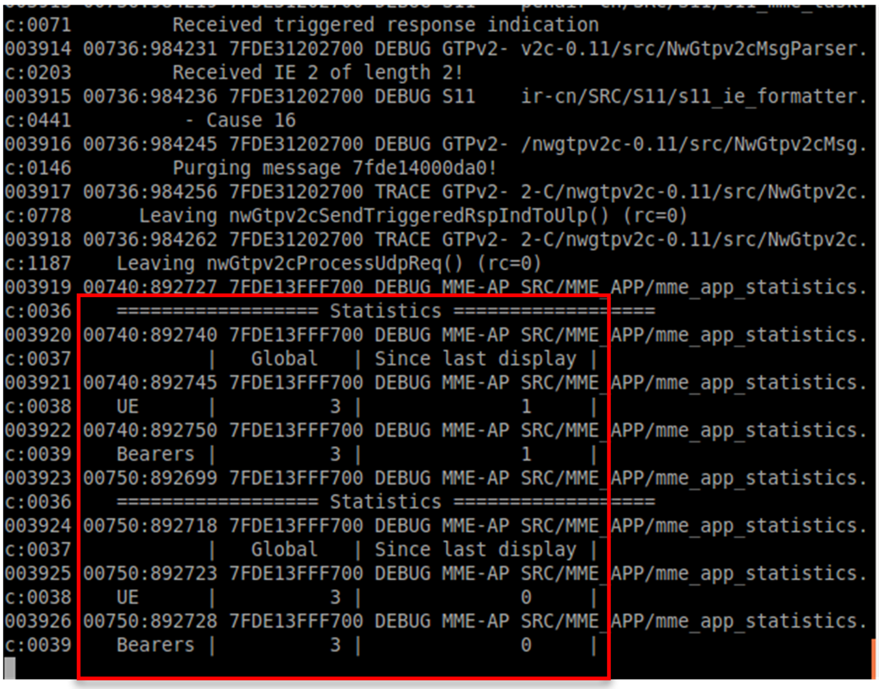

LTE UL SCOPE eNB for CC_id 0, UE 1
6:23 PM
LTE UL SCOPE eNB for CC_id 0, UE 0
Received Signal (Time-Domain, dB)
Channel Impulse Response (samples, abs)
Channel Frequency Response (RE, dB)
PUSCH Log-Likelihood Ratios (LLR, mag)
PUSCH I/Q of MF Output
PUSCH Throughput [frame]/[kbit/s]
PUCCH1 Energy (SR)
PUCCH I/Q of MF Output
DL Traffic OFF

```
root@flexicell2: /OAI_CPRI/openairinterface5g 115x61
            </InterFreqBandInfo>
          </interFreqBandList>
        </BandInfoEUTRA>
[RRC][I][FRAME 44195][eNB][MOD 00][RNTI a80d] UE State = RRC_RECONFIGURED
[SCTP][I][sctp_send_data] Successfully sent 40 bytes on stream 1 for assoc_id 44
[PHY][I][eNB 0] Frame 112: Sent physicalConfigDedicated=0x7f516800a6d0 for UE 1
[RRC][N][eNB 0] Frame 114: received a DCCH 1 message on SRB 0 with Size 16 from UE a80d
[RRC][I][FRAME 44197][eNB][MOD 00][RNTI a80d] Received on DCCH 1 RRC_DCCH_DATA_IND
[SCTP][I][sctp_send_data] Successfully sent 61 bytes on stream 1 for assoc_id 44
[SCTP][I][sctp_eNB_flush_sockets] Found data for descriptor 38
[SCTP][I][sctp_eNB_read_from_socket] Received notification for sd 38, type 32777
```

```
flexicell2@flexicell2: /OAI_CPRI/openairinterface5g 115x61
flexicell2@flexicell2:/OAI_CPRI/openairinterface5g$ cd /OAI_CPRI/openairinterface5g/
flexicell2@flexicell2:/OAI_CPRI/openairinterface5g$ ./restoreCpi
[sudo] password for flexicell2:
```

Figure 7.20: 2 UEs successfully connected to the BBU.

Although the OAI LTE soft-modem can only display two uplink scopes, tests with up to three UEs were conducted successfully. Figure 7.21 shows the output of the MME process running in the EPC. 3 UEs and respective bearers can be seen simultaneously attached.



```
c:0071          Received triggered response indication
003914 00736:984231 7FDE31202700 DEBUG GTPv2- v2c-0.11/src/NwGtpv2cMsgParser.
c:0203          Received IE 2 of length 2!
003915 00736:984236 7FDE31202700 DEBUG S11    ir-cn/SRC/S11/s11_ie_formatter.
c:0441           - Cause 16
003916 00736:984245 7FDE31202700 DEBUG GTPv2- /nwgtpv2c-0.11/src/NwGtpv2cMsg.
c:0146          Purging message 7fde14000da0!
003917 00736:984256 7FDE31202700 TRACE GTPv2- 2-C/nwgtpv2c-0.11/src/NwGtpv2c.
c:0778       Leaving nwGtpv2cSendTriggeredRspIndToUlp() (rc=0)
003918 00736:984262 7FDE31202700 TRACE GTPv2- 2-C/nwgtpv2c-0.11/src/NwGtpv2c.
c:1187     Leaving nwGtpv2cProcessUdpReq() (rc=0)
003919 00740:892727 7FDE13FFF700 DEBUG MME-AP SRC/MME_APP/mme_app_statistics.
c:0036     ================== Statistics ==================
003920 00740:892740 7FDE13FFF700 DEBUG MME-AP SRC/MME_APP/mme_app_statistics.
c:0037            |   Global    | Since last display |
003921 00740:892745 7FDE13FFF700 DEBUG MME-AP SRC/MME_APP/mme_app_statistics.
c:0038     UE     |           3 |          1         |
003922 00740:892750 7FDE13FFF700 DEBUG MME-AP SRC/MME_APP/mme_app_statistics.
c:0039     Bearers |          3 |          1         |
003923 00750:892699 7FDE13FFF700 DEBUG MME-AP SRC/MME_APP/mme_app_statistics.
c:0036     ================== Statistics ==================
003924 00750:892718 7FDE13FFF700 DEBUG MME-AP SRC/MME_APP/mme_app_statistics.
c:0037            |   Global    | Since last display |
003925 00750:892723 7FDE13FFF700 DEBUG MME-AP SRC/MME_APP/mme_app_statistics.
c:0038     UE     |           3 |          0         |
003926 00750:892728 7FDE13FFF700 DEBUG MME-AP SRC/MME_APP/mme_app_statistics.
c:0039     Bearers |          3 |          0         |
```

Figure 7.21: 3 UEs successfully attached to the EPC.

These tests were performed with a smartphone (Samsung Galaxy S4 Active), USB dongle (Huawei E398 and/or E3276) and 4G LTE router (TP-LINK M7350) connected simultaneously, proving functional connectivity with several commercial off-the-shelf devices.

CHAPTER 8

Conclusion and Future Work

In this chapter conclusions are highlighted followed by proposed future lines of work.

8.1 Conclusions

Within the scope of this dissertation, a successful proof of concept was achieved regarding an Ethernet-based C-RAN fronthaul implementation and validated in a fully working C-RAN test-bed with a 20 km optical link.

C-RAN architectures with the underlying concept of centralized and cooperative processing and widespread deployment of simple remote radio heads could solve MNOs current coverage and capacity problems for the upcoming 5G in a sustainable way. Although C-RAN brings numerous advantages, it also exerts great stress on the fronthaul that traditional solutions cannot cope with. Furthermore, currently used CPRI-based fronthauls are intrinsically inflexible and inefficient. As such, development of a 10 Gigabit Ethernet fronthaul solution was proposed for being more flexible, efficient and allowing easier integration with existing industry standard infrastructure and equipment.

A working LTE C-RAN test-bed previously developed at Instituto de Telecomunicações - Aveiro, served as the basis for the framework. This C-RAN test-bed was originally developed with a 20 km CPRI-based fronthaul and this dissertation main objective was to replace it with a 10 Gigabit Ethernet fronthaul. Both original project and the one developed in this dissertation were done on top of FPGA-based development kits by Xilinx and with the support of OpenAirInterface open-source software for the mobile network intelligence.

Earlier work stages involved the study and familiarization of the original setup. Those include the hardware infrastructure, FPGA digital designs, embedded SDK application and the use of the OpenAirInterface platform (LTE soft-modem + EPC).

Moving to the 10 Gigabit Ethernet fronthaul implementation, a standalone 10 Gigabit Ethernet link was developed on top of two KC705 development kits. Both hardware and software are developed on Xilinx's development environments - Vivado Design Suite and SDK.

Hardware development was based on a mix of IP cores, block design and VHDL/Verilog hardware description languages.

The 10G Ethernet solution is based on Xilinx's 10G Ethernet Subsystem IP core that already implements the 10G Ethernet MAC and PHY (PCS/PMA). The lower PHY layer (PMD) was implemented through optical SFP transceivers. Xilinx provides a simple example design with the IP core that serve as a basis for the implementation. In a first instance the developed solution is validated with two KC705 boards exchanging frames with each other. After the standalone 10G Ethernet link is validated, the necessary adjustments are made in order to integrate the 10G Ethernet design with the original C-RAN design. Integration presented several difficulties, most of them related to the clocking and synchronization (or lack of) aspects of the design, asserting the importance of synchronization concerns shown in introductory chapters.

By the end of the integration, the Ethernet-based C-RAN test-bed was validated successfully using the 20 km optical fronthaul with multiple user equipments connected simultaneously to the network and with Internet access. Although the test-bed is now working with separate and dedicated channels for user data, control and synchronization planes, the proof of concept is considered to have been proven.

Also to be considered, the setup now features a 10Gbit Ethernet point-to-point link without any switching equipment in between. The introduction of that kind of equipment and the added non-determinism of the packets and extra latency means that buffering might be needed to offset this problems. As this buffering will further increase the latency, the fronthaul link might likely suffer a reduction on maximum supported range.

8.2 Future Work

Thanks to the combined use of FPGA-based development kits, SDR platforms like the AD9361 RF transceiver and OpenAirInterface open-source software, the current C-RAN test-bed is an extremely open and flexible setup. As such, multiple lines of work are possible:

- Achieve synchronization by implementing SyncE and/or IEEE 1588v2 PTP synchronization protocols through the fronthaul optical link;
- Merge the control and management plane with the data plane through the fronthaul optical link;
- Introduce switching equipment in the fronthaul link to evaluate the coexistence with other network equipment;
- Explore other function splittings to reduce required fronthaul bandwidth by implementing low level functions such as FFT/IFFT on the FPGAs;
- Increase the number of RRHs connected to the BBU.

Appendices

APPENDIX A

Flexicell Start-Up User Guide

Detailed user guide on how to successfully start-up the original CPRI-based laboratory C-RAN test-bed.

A.1 RRH

(a) Boot the RRH support computer (labelled Flexicell 1), Ubuntu 14.04.5 LTS (on /dev/sda3).

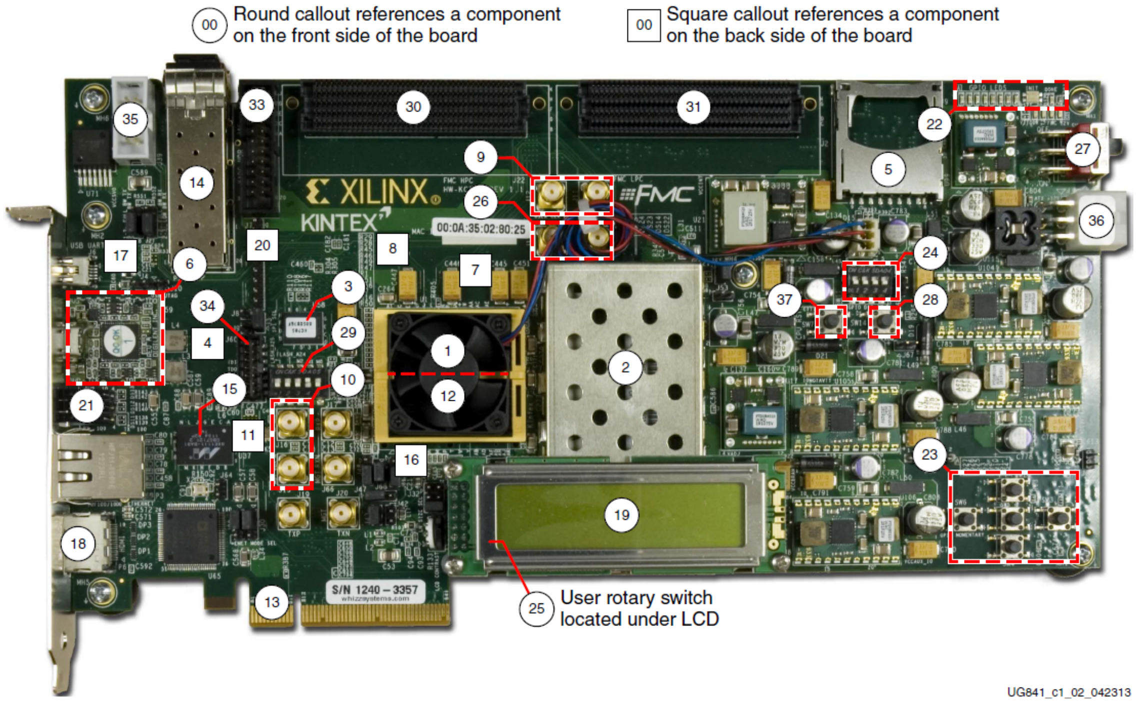

00 Round callout references a component on the front side of the board
00 Square callout references a component on the back side of the board
35
33
30
31
22
27
14
9
5
26
36
20
8
17
6
7
24
3
34
37
28
29
4
1
2
10
15
12
21
11
16
23
19
18
13
S/N 1240 – 3357
25 User rotary switch located under LCD
XILINX
KINTEX
FMC
00:0A:35:02:80:25
UG841_c1_02_042313

Figure A.1: KC705 Board Components (Retrieved from [Xil16d]).

(b) Connect a Micro USB cable to the USB-JTAG connector (6) in the KC705 board and a Type B USB cable to the CDCDE72010EVM Texas Instruments evaluation module. If pretended to use the UART, connect a Mini USB cable to the KC705 USB-to-UART connector (17).
(c) Ensure that the optical fibers / SFP+ module are connected in the SFP cage(14).
(d) Turn On the KC705 board (27).
(e) Start Oracle VirtualBox and open the Windows VM.
(f) Allow the CDCDE72010EVM module and the UART bridge to be accessed by the VM. These should appear as "**Texas Instruments, Inc.**" and "**Silicon Labs CP2103 USB-to-UART Bridge Controller**".
(g) Open the **CDCDE72010 GUI** software and load the configuration file by selecting: **File** → **Open INI** → **flexicell_cpe_re_rrh_pll_config.ini**.
(h) Launch Vivado 2014.4.
(i) Open the RRH project: **cpe_re_rrh_comp**.
(j) Select **File** → **Launch SDK**.
(k) In the SDK Application, select **Xilinx Tools** → **Program FPGA** → **Program**.
(l) On the left, right-click on **cpe_re_rrh_sw** → **Run As** → **Launch On Hardware**.

A.2 EPC

(a) Still on the RRH support computer, open 3 terminal windows. In each one of them: connect to the EPC via SSH → ***ssh it@striker*** → ***sudo su*** → ***cd /openair-cn/SCRIPTS***.
(b) Run the ***./run_spgw*** script in the first terminal, ***./run_hss*** in the second terminal and ***./run_mme*** in the last terminal.

A.3 BBU

(a) Turn On the support laptop and connect the Si570 Evaluation Board with a Type B USB cable.
(b) Open the **Programmable Oscillator Calculator** software.
(c) Select **Si570** and **Connect to EVB** → OK.
(d) Insert start-up frequency of **156,25 MHz** → **Apply Definition**.
(e) Insert new frequency of **307,2 MHz** → **Run Procedure**.
(f) Verify in the bottom of the window that the device is programmed successfully → "**Program Complete!**".

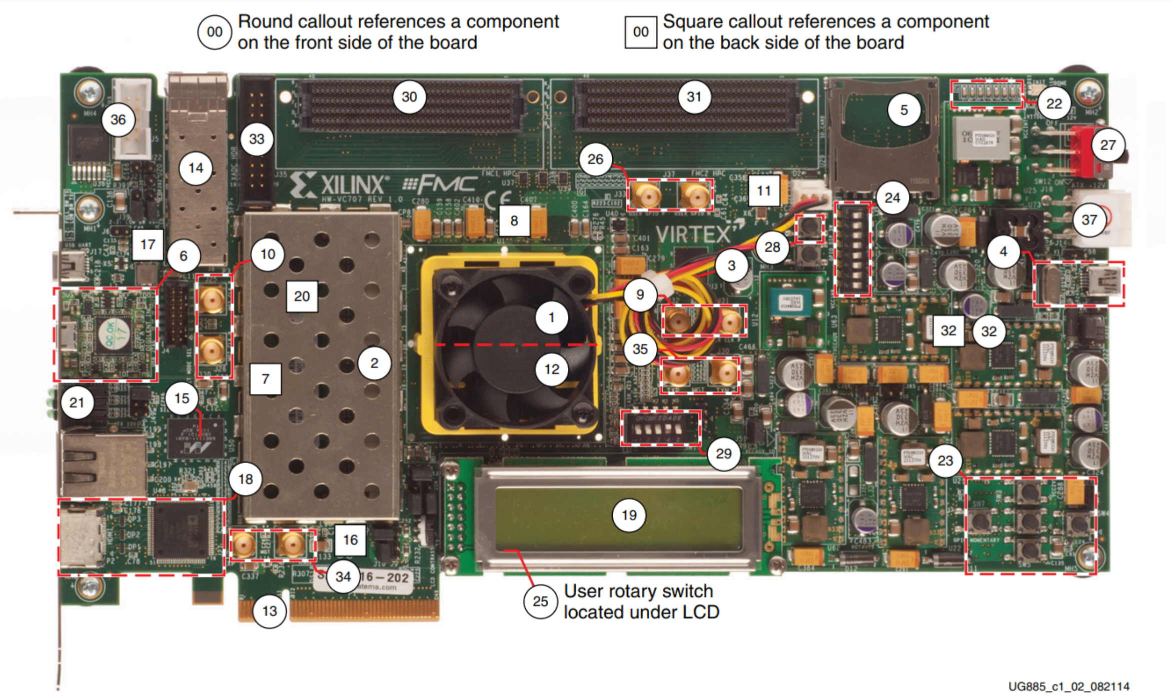

00 Round callout references a component on the front side of the board
00 Square callout references a component on the back side of the board
30
31
36
33
14
26
5
22
27
11
24
8
37
17
6
10
3
28
4
20
9
1
32
32
2
35
12
7
21
15
29
18
23
19
16
34
13
25 User rotary switch located under LCD
UG885_c1_02_082114

Figure A.2: VC707 Board Components (Retrieved from [Xil16e]).

(g) Connect a Micro USB cable to the USB-JTAG connector (6) in the VC707 board. If pretended to use the UART, connect a Mini USB cable to the VC707 USB-to-UART connector (17).

(h) Ensure that the optical fibers / SFP+ module are connected to the native SFP cage(14).

(i) Still in the VC707 board, User DIP Switch (24), put the first switch in position 1.

(j) Turn on the VC707 board (37).

(k) Boot the BBU computer (labelled Flexicell 2), Ubuntu 14.04 3.19.0-61-lowlatency.

(l) Launch Vivado 2014.4.

(m) Open the BBU project: **vc707_pcie_cpri_comp**.

(n) Select **File** → **Launch SDK**.

(o) In the SDK Application, select **Xilinx Tools** → **Program FPGA** → **Program**.

(p) Verify the **User LEDs** (22) in both boards. They should read **X1001010**. X - Blinking, 1 - ON, 0 - OFF.

(q) Reboot the BBU computer and redo steps (k) to (n).

(r) Open a terminal window, change directory to → ***cd /OAI_CPRI/openairinterface5g***.

(s) Run the **saveCPi** script → ***./saveCpi***.

(t) Back at the SDK Application, on the left, right-click on **pcie** → **Run As** → **Launch On Hardware**.

(u) Run the **restoreCPi** script → ***./restoreCpi***.

(v) Lastly run the ***./run_Flexicell*** script and the RAN should be operational.

APPENDIX B

OpenAirInterface Deployment

OpenAirInterface EPC and BBU (eNB) deployment.

The deployment begins by installing a Linux distribution onto the hard-drives, in this case Ubuntu 14.04 for both BBU and EPC computers. The EPC is installed on a virtual machine running on a rack-mounted server and don't requires any special requirements. In the case of the BBU, the LTE soft-modem runs directly on the physical machine and requires a low-latency kernel and that all processor frequency scaling and power management features are disabled. This means that C-States, P-States and Turbo Boost technologies are deactivated either in the BIOS or configuration files inside the operating system. Furthermore, OAI also recommends to disable hyper-threading if the CPU support the technology. These requirements are necessary to avoid real-time issues between the BBU and RRH/UEs [Ope17].

After previous preparations are completed, follows installation and configuration of OAI eNB (BBU) and EPC on respective host machines. The open source software is retrieved from the "openairinterface5g" repository at Gitlab using Git. In the BBU case, along with the LTE soft-modem, drivers for the RF equipment are also installed. In this case, since it is being implemented a C-RAN architecture, the RF equipment won't be connected directly to the BBU as in OAI original setup, as such, drivers from the EXMIMO RF board are installed but later replaced by custom drivers to allow interfacing with the VC707 development board that implements the fronthaul interface with the BBU. From the EPC side, the open source software is retrieved from the "openair-cn" git repository and SPGW, HSS and MME are installed separately. Finished the installation, changes on configuration files on both hosts are necessary. This changes include network interfaces configuration (IP addresses, ports, MTU, hostname, etc) and RAN parameters such as MNC, MCC, etc. Lastly, SIM cards information needs to be registered in the HSS MySQL database in order for the UEs to connected and register properly in the network.

References

[AB+15] Ericsson AB et al. *Common Public Radio Interface (CPRI): Interface Specification.* CPRI Specification V7.0 (2015-10-09). 2015.

[AB+18a] Ericsson AB et al. *Common Public Radio Interface: eCPRI Interface Specification.* eCPRI Specification V1.1 (2018-01-10). 2018.

[AB+18b] Ericsson AB et al. *Common Public Radio Interface: Requirements for the eCPRI Transport Network.* eCPRI Transport Network V1.1 (2018-01-10). 2018.

[Ana17a] AnalogDevices. *AD-FMCOMMS3-EBZ User Guide.* 2017. URL: `https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms3-ebz` (Accessed: October 24, 2017).

[Ana17b] AnalogDevices. *RF Agile Transceiver.* 2017. URL: `http://www.analog.com/en/products/rf-microwave/integrated-transceivers-transmitters-receivers/wideband-transceivers-ic/ad9361.html` (Accessed: October 24, 2017).

[Anj+15] Gustavo Anjos et al. "Implementation and Evaluation of a Low Latency and Resource Efficient Compression Method for Digital Radio Transport of OFDM Signals". In: *2015 IEEE Globecom Workshops (GC Wkshps)* (2015). DOI: `10.1109/glocomw.2015.7413994`.

[Bar+17] J. Bartelt et al. "5G transport network requirements for the next generation fronthaul interface". In: *Eurasip Journal on Wireless Communications and Networking* 2017.1 (2017). ISSN: 16871499. DOI: `10.1186/s13638-017-0874-7`.

[Cis17] Cisco. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021.* White Paper. February 7, 2017.

[Fin09a] Finisar Corporation. *1G/10G 850nm Multimode Datacom SFP+ Transceiver.* Product Specification, REV. B. 2009.

[Fin09b] Finisar Corporation. *Product Specification, DWDM SFP Transceiver.* FWLF1631xx. 2009.

[Fre13] Lou Frenzel. "Small-Cell And HetNet Movement". In: *Electronic Design* 61.11 (2013), 48(8).

[Gom+15] Nathan J. Gomes et al. "Fronthaul evolution: From CPRI to Ethernet". In: *Optical Fiber Technology* 26 (2015), pp. 50–58. ISSN: 10685200. DOI: `10.1016/j.yofte.2015.07.009`.

[Har14] S.M. Shin Harrison J. Son. *Fronthaul Size: Calculation of maximum distance between RRH and BBU.* 2014. URL: `https://www.netmanias.com/en/?m=view&id=blog&no=6276` (Accessed: June 14, 2018).

[I+15] Chih-lin I et al. "Rethink fronthaul for soft RAN". In: *IEEE Communications Magazine* 53.9 (2015), pp. 82–88. DOI: `10.1109/mcom.2015.7263350`.

[Ins13] China Mobile Research Institute. *"C-RAN The Road Towards Green RAN", White Paper, Version 3.0 (Dec, 2013).* 2013.

[Ins15] China Mobile Research Institute. *White Paper of Next Generation Fronthaul Interface.* Version 1.0. 2015.

[Lab14a] Silicon Labs. *Si514/570/571/598/599 Any-Frequency I2C Programmable XO/VCXO Evaluation Board.* Rev 0.3. 2014.

[Lab14b] Silicon Labs. *Si570/Si571 10 MHz To 1.4 GHz I2C Programmable XO/VCXO.* Rev 1.5. 2014.

[Lan+16] Christian Lanzani et al. *Fronthaul Evolution Toward 5G: Standards and Proof of Concepts.* 2016. URL: `https://www.design-reuse.com/articles/40008/fronthaul-evolution-toward-5g-standards-and-proof-of-concepts.html` (Accessed: August 30, 2016).

[Oli16] Arnaldo S. R. Oliveira. *FLEXICELL C-RAN Lab Testbed.* September, 2016.

[Ope17] OpenAirInterface. *Towards Open Cellular Ecosystem.* 2017. URL: `http://www.openairinterface.org/?page_id=864` (Accessed: October 24, 2017).

[Ora+] Orange et al. *Fronthaul Cloud RAN Evolutions.* White Paper.

[Pen+15] Mugen Peng et al. "Fronthaul-constrained cloud radio access networks: insights and challenges". In: *IEEE Wireless Communications* 22.2 (2015), pp. 152–160. DOI: `10.1109/mwc.2015.7096298`.

[Pfe15] Thomas Pfeiffer. "Next Generation Mobile Fronthaul and Midhaul Architectures [Invited]". In: *Journal of Optical Communications and Networking* 7.11 (2015), B38. DOI: `10.1364/jocn.7.000b38`.

[Qua14] Qualcomm. *"The Evolution of Mobile Technologies".* June, 2014.

[Qua16] Qualcomm Technologies, Inc. *Making 5G NR a reality.* Leading the technology innovations for a unified, more capable 5G air interface. 2016.

[San+16a] Jorge Santos et al. "A Flexible Physical Layer and Fronthaul Research Testbed for C-RAN". In: (2016).

[San+16b] Jorge Santos et al. *Flexicell – OAI-based Cloud Radio Access Networks.* 2016. URL: `http://www.openairinterface.org/?page_id=1638` (Accessed: October 24, 2017).

[SFF09] SFF Committee. *SFF-8431 Specifications for Enhanced Small Form Factor Pluggable Module SFP+.* 2009.

[Tec14] Faster Technology. *FM-S14 User Manual.* Quad SFP/SFP+ transceiver FMC. 2014.

[Tex08] Texas Instruments. *1.5-GHz Low-Phase Noise Clock Evaluation Board.* User Guide. 2008.

[Xil13] Xilinx. *Xilinx 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide for HDL Designs.* UG768 (v14.7). 2013.

[Xil14a] Xilinx. *AR 59750, Kintex-7 FPGA KC705 Evaluation Kit - Changes from rev 1.0 to rev 1.1.* 2014. URL: `https://www.xilinx.com/support/answers/59750.html` (Accessed: September 3, 2017).

[Xil14b] Xilinx. *Integrated Logic Analyzer v5.0 LogiCORE IP Product Guide.* PG172. 2014.

[Xil14c] Xilinx. *Vivado Design Suite User Guide Implementation.* UG904 (v2014.4). 2014.

[Xil15] Xilinx. *FIFO Generator.* v12.0. PG057 June 24, 2015.

[Xil16a] Xilinx. *10 Gigabit Ethernet Subsystem.* v3.1. PG157 October 5, 2016.

[Xil16b] Xilinx. *AXI 1G/2.5G Ethernet Subsystem.* v3.1. PG157 October 5, 2016.

[Xil16c] Xilinx. *7 Series FPGAs GTX/GTH Transceivers User Guide.* UG476 (v1.12). 2016.

[Xil16d] Xilinx. *KC705 Evaluation Board for the Kintex-7 FPGA User Guide.* UG810 (v1.7). 2016.

[Xil16e] Xilinx. *VC707 Evaluation Board for the Virtex-7 FPGA User Guide.* UG885 (v1.7.1). 2016.

[Xil16f] Xilinx. *Vivado Design Suite User Guide Programming and Debugging.* UG908 (v2016.1). 2016.

[Xil17] Xilinx. *High Speed Serial.* 2017. URL: `https://www.xilinx.com/products/technology/high-speed-serial.html` (Accessed: October 6, 2017).

[Xil18a] Xilinx. *7 Series FPGAs SelectIO Resources User Guide.* UG471 (v1.10). 2018.

[Xil18b] Xilinx. *Xilinx Kintex-7 FPGA KC705 Evaluation Kit.* 2018. URL: `https://www.xilinx.com/products/boards-and-kits/ek-k7-kc705-g.html#hardware` (Accessed: July 3, 2018).